Comparative Study of the L_1 Norm Regression Algorithms

Bijan Bidabad[1]

Keywords: L_1 norm, Regression, Algorithm, Computer

Abstract:

This paper tries to compare more accurate and efficient L_1 norm regression algorithms. Other comparative studies are mentioned and their conclusions are discussed. Many experiments have been performed to evaluate the comparative efficiency and accuracy of the selected algorithms.

1. Introduction

The objective of this paper is to compare some of the existing algorithms for the L_1 norm regression with those proposed by Bidabad (1989a,b). Our point of view is to compare the accuracy and relative efficiencies of them. In this respect accuracy of the solution of the algorithms is more important than the other criteria. By the term accuracy we mean, reaching the correct solution in finite number of steps or iterations. By efficiency, we mean that the algorithm performs with smaller amount of required storage and execution time to reach accurate optimal solution.

Generally, the comparison of algorithms is not a straightforward task. As it is indicated by Dutter (1977), factors such as quality of computer codes and computing environment should be considered. In the case of the L_1 norm algorithms, three specific factors of number of observations, number of parameters and the condition of data are more important. Kennedy and Gentle and Sposito (1977a,b), and Hoffman and Shier (1980a,b) describe methods for generating random test data with known L_1 norm solution vectors. Gilsinn et al (1977) discuss a general methodology for comparing the L_1 norm algorithms. Kennedy and Gentle (1977) examine the rounding error of L_1 norm regression and present two techniques for detecting inaccuracies of the computation (see also, Larson and Sameh (1980)).

Many authors have compared their own algorithms with those already proposed. Table 1 gives a summary of the characteristics of the algorithms proposed by different authors. It is important to note that since the computing environment and condition of data with respect to distribution of the regression errors of the presented algorithms by table 1 are not the same, definitive conclusion and comparison should not be drawn from this table.

Armstrong and Frome (1976a) compare the iterative weighted least squares of Schlossmacher (1973) with Barrodale and Roberts (1973) algorithm. The result was high superiority of the latter. Anderson and Steiger (1980) compare the algorithms of Bloomfield and Steiger (1980), Bartels and Conn and Sinclair (1978) and Barrodale and Roberts (1973). It was concluded that as number of observations n increases the BR locates in a different complexity class than BCS and BS. All algorithms are linear in number of parameters m, and BS is less complex than BCS. Complexities of BS and BCS are linear in n. There is a slight tendency for

[1] Research professor of economics, Monetary and Banking Research Academy, Central Bank of Iran, Pasdaran, Zarabkhaneh, Tehran, 16619, Iran. bijan_bidabad@msn.com.

I should express my sincere thanks to Professor Yadolah Dodge from Neuchatel University of Switzerland who taught me many things about L_1 norm when he was my Ph.D.dissertation advisor.

all algorithms to work proportionately harder for even m than for odd m. BR and BS had the most difficulty with normal error distribution and the least difficulty with Pareto distribution with corresponding Pareto density parameter equal to 1.2.

Table 1. Summary of the characteristics of the existing algorithms.

ref.	Compared with	m range	n range	Time/performance
BCS	BR	2-8	201	roughly equal speed
AFK	BR	5-20	100-1500	30%-50% AFK is faster
A	BR	1-11	15-203	nearly equal speed
BS	BR	2-6	100-1800	BS is faster for larger n
W	AFK,AK	2-25	100-1000	W is faster for larger n & smaller m
SS	BS	4-34	10-50	SS is faster for m near n
AK	S	2	50-500	AK is faster
JS	Ak	2	10-250	JS is faster

n ≡number of observations.
m ≡number of parameters.
BCS≡Bartels,Conn,Sinclair(1978).
BR ≡Barrodale,Roberts(1973,74).
AK ≡Armstrong,Kung(1978).
S ≡Sadovski(1974).
AFK≡Armstrong,Frome,Kung(1979).
A ≡Abdelmalek(1980a,b).
BS ≡Bloomfield,Steiger(1980).
W ≡Wesolowsky(1981).
JS ≡Josvanger,Sposito(1983).
SS ≡Seneta,Steiger(1984).

Gentle and Narula and Sposito (1987) performs a rather complete comparison among some of the L_1 norm algorithms. They limited this comparison to the codes that are openly available for L_1 norm linear regression of unconstrained form. Table 2 shows the required array storage and stopping constants of the corresponding algorithms.

Table 2. Array storage requirement for selected algorithms.

Program name	Ref.	required array storage	stopping constants

L_1	BR	3n+m(n+5)+4	BIG=1.0E+75
			TOLER=10**(-D+2/3)
			D=No. of decimal digits of accuracy

L_1	A	6n+m(n+3m/2+15/2)	PREC=1.0E-6
			ESP=1.0E-4

L_1NORM	AFK	6n+m(n+m+5)	ACU=1.0E-6
			BIG=1.0E+15

BLAD1	BS	4n+2m(n+2)	--------------------

LONESL	S	4n	PREC=1.0E-6
			BIG=1.0E+19

SIMLP	AK	4n	ACU=1.0E-6
			BIG=1.0E+19

DESL$_1$	JS	5n	TOL=1.0E-6

See table 1 for abbreviations.
Source: Gentle, Narula, Sposito (1987).

In their study the problem consists of uniform (0,1) random values for X and normal (0,3) variates for random error term. The value of dependent variable y computed as sum of the independent variables and error term. Summary of the results is shown in tables 3 and 4 for simple and multiple regressions respectively. Values in the cells are the CPU time averages of 100 replications and the values in the parentheses are corresponding maximum CPU time of the 100 replications. Gentle and Sposito and Narula (1988) also compare the algorithms for unconstrained L_1 norm simple linear regression. This investigation is essentially an extraction of Gentle and Narula and Sposito (1987). The attained results are completely similar.

They concluded that BS program performs quite well on smaller problems but in larger cases, because of accumulated round-off error it fails to produce correct answers. The Wesolowsky program was not usable and deleted in their study. Because of superiority of AFK to BR and AK to S which had been indicated in previous studies, BR and S algorithm did not enter in their study.

Table 3. CPU time for simple model.

n	AK	JS	A	AFK	BS
100	0.021	0.023	0.094	0.034	0.023
	(0.03)	(0.04)	(0.21)	(0.06)	(0.04)
500	0.193	0.302	1.434	0.287	0.145
	(0.38)	(0.61)	(3.13)	(0.49)	(0.26)
1000	0.544	0.971	4.775	0.784	0.422
	(1.36)	(2.16)	(10.60)	(1.76)	(1.19)
5000	1.262	2.837	211.23*	1.614	+
	(24.58)	(48.88)	(----)	(31.22)	+

See table 1 for abbreviations.
* Average of three runs.
+ Failed to produce correct answers.
Source: Gentle, Narula, Sposito (1987).

Table 4. CPU time for multiple model (m=5,15).

n	m	A		AFK		BS	
100	5	0.331	(0.53)	0.149	(0.23)	0.114	(0.17)
100	15	1.976	(2.73)	1.313	(1.70)	0.933	(1.38)
500	5	3.686	(5.47)	1.120	(1.81)	0.829	(1.22)
500	15	17.876	(23.4)	7.808	(10.1)	7.294	(9.13)
1000	5	13.211	(18.3)	2.930	(4.38)	+	+
1000	15	49.866	(72.7)	17.901	(24.0)	+	+
5000	5	248.91*	(----)	34.311	(51.8)	+	+
5000	15	687.31*	(----)	140.321	(160.1)	+	+

See table 1 for abbreviations.
* Average of three runs.
+ Failed to produce correct answers.
Source: Gentle, Narula, Sposito (1987).

By considering all aspects, they concluded that AFK seems to be the best.

2. Design of experiments

Performance of every algorithm in any specific computing environment is different and thus makes the absolute comparison of algorithms very difficult, especially if the system uses, virtual or real storage, a cache or any array processor or mathematical co-processor and etc. As it was discussed by Bidabad (1989a,b), many algorithms exist for L_1 norm regression with corresponding computer program and comparison of all of them is very costly. In order to reduce the number of experiments, we rely on the experience of previous researchers which were discussed above. However, the experiments are divided into two general categories of simple and multiple linear L_1 norm regressions.

Despite of the coded computer programs, computing environment, numbers of observations and parameters of the model and "condition" of data are the major sources of comparisons for performances of algorithms. Thus different sizes problems are to be tested in this section.

To judge the superiority of algorithms there are many criteria. Accuracy and efficiency are basic ones. In the former, we are concerned with obtaining the true results in different samples and in the latter the computation time and storage requirement of the algorithms are compared.

To perform the experiments, once uniform random values selected for ßj in the following model,

$$y_i = \sum_{j=1}^{m} \beta_j x_{ij} + u_i \qquad i=1,...,n \tag{1}$$

Random values generated for x_{ij} and u_i with five specifications of distributions. Uniform and normal random generators (given by Mojarrad (1977)) used to generate three uniform and two normal sets of random data for each experiment. Generated uniform random deviates belong to the [-10,10], [-100,100] and [-1000,1000] intervals. Normal deviates have zero mean with 100 and 1000 variances. Values of y_i computed for $ß_j$, x_{ij}, and u_i which had been generated as explained above. Values of 20, 50, 100, 500, 1000, 2000, 5000 and 10000 were used for number of observations n and values of 2, 3, 4, 5, 7 and 10 selected for number of parameters m.

Hence, for all of the five specifications of distribution of u_i, and for all m and n, a replication is done for each of the selected algorithms. Average and range of these five replications are reported for each m and n for each algorithm. In the case of simple regression number of replications is ten than five.

The programs were all complied by Fortran IV, VS compiler, 1.3.0 level (May 1983) and 77 LANGLVL with 03 optimization level to reduce the coding inefficiencies. The programs were run on BASF 7.68 (MVS) computer. Since this machine is a multitasking system, swapping process should affect the execution time. When system is running for more than one job, this effect can not be measured and removed completely. In order to filter the swapping time, Service Request Block (SRB) time has been reduced from the total Central Processing Unit (CPU) time. However, when system is busy this may not exhaust all the swapping times. It has been tried to run all comparable algorithms simultaneously, and also in one class of input with enough initiators and the same priority level to cause similar situations for all comparable submitted jobs. The pre-execution times of compilation and linkage-editor are excluded from all tested programs.

3. Comparison of the simple regression L_1 norm algorithms

In this study comparisons are limited to the algorithm 2 of Bidabad (1989a,b) and that proposed by Josvanger and Sposito (1983). Gentle and Narula and Sposito (1987) and Gentle and Sposito and Narula (1988) introduced the latter as the most efficient algorithm for simple linear L_1 norm regression.

Table 5. Array storage requirement for simple model selected algorithms

Algorithm	Program name	Storage requirement	Stopping constant
Js	DESL$_1$	5n	TOL = 1.0E-6
B (Alg.2)	BL$_1$S	5n	------------

Js ≡ Josvanger and Sposito (1983).
B(Alg.2) ≡ Bidabad (1989a,b) Algorithm 2.
n ≡ Number of observations.

The amount of array storage requirement for these two programs are shown by table 5. This table may be compared with table 2 for other algorithms. None of the programs destroys the input data. Both programs have been coded in single precision.

Table 6 shows the results of the experiments for simple linear L_1 norm regression. The values reported in the cells of the table are the averages of ten replications CPU times in seconds with different random samples. The values in the parentheses are the corresponding minimum and maximum CPU times of the ten runs. Both algorithms converged and gave accurate results for all of the experiments.

As it is clear from table 6 in small samples the computation times are not very different, though algorithm 2 is faster. In medium samples, this difference becomes significant and in larger samples, algorithm 2 becomes strongly superior to that of Josvanger and Sposito (1983).

Thus it can be concluded that algorithm 2 performs better than the other algorithms and may be used for applied work to achieve more efficiency.

Table 6. CPU times for simple model

	L_1 norm selected algorithms	
n	JS	B(Alg.2)
20	0.096	0.094
	(0.09,0.10)	(0.09,0.10)
50	0.109	0.106
	(0.10,0.11)	(0.10,0.11)
100	0.141	0.132
	(0.13,0.16)	(0.12,0.14)
500	0.469	0.360
	(0.35,0.59)	(0.33,0.38)
1000	0.997	0.645
	(0.66,1.31)	(0.61,0.69)
2000	2.770	1.194
	(1.36,4.27)	(0.98,1.28)
5000	11.554	2.848
	(4.58,18.91)	(2.71,3.15)
10000	42.406	5.823
	(9.67,60.88)	(5.16,6.87)

See table 5 for abbreviations.

4. Comparison of the multiple regression L_1 norm algorithms

To compare algorithm 4 of Bidabad (1989a,b) with other algorithms, experiments have been limited to three algorithms which are more accurate and efficient among the others. These are algorithms of Barrodale and Roberts (1973,74) (BR), Bloomfield and Steiger (1980) (BS), Armstrong and Frome and Kung (1979) (AFK). Although, BS and AFK algorithms are faster than BR, the reason to select BR algorithm was that the other two algorithms failed to produce correct answers for larger samples (see, Gentle and Narula and Sposito (1987)).

The amount of array storage requirement for these programs are indicated by table 7. This table may be compared with table 2 for other algorithms. All programs have been coded in single precision. None of the programs destroys input data.

Table 7. Array storage requirement for multiple model selected algorithms

Algorithm	Program name	Storage requirement	Stopping constant

AFK	AFKL$_1$	6n+m(n+3m/2+15/2)	ACU = 1.0E-6
			BIG = 1.0E+15

BR	L$_1$BAR	3n+m(n+5)+4	BIG = 1.0E+75
BS	BLOD1	4n+m(2n+4)	-------------
B (Alg.4)	BL$_1$	2n+m(3n+m+2)-2	-------------

AFK ≡ Armstrong and Frome and Kung (1970).
BR ≡ Barrodale and Roberts (1973,74).
BS ≡ Bloomfield and Steiger (1980).
B(Alg.4) ≡ Bidabad (1989a,b) Algorithm 4.
n ≡ Number of observations.
m ≡ Number of parameters.

Tables 8 through 12 report the averages of five runs CPU times for different sample sizes and parameters. The values in the parentheses are minimum and maximum CPU times of replications. For the three parameters model, as it can be seen from table 8, the algorithm 4 is superior to other algorithms. In this case the BS, AFK, and BR possess less efficiency respectively. When the sample size is small, the difference is not large. In medium sample sizes this difference is going to increase. In larger size experiments, algorithm 4 and BS have small difference, but BR and AFK are far from them. In all cases algorithm 4 is faster than the other algorithms.

Table 8. CPU times for multiple model (m=3) selected algorithms

n	B(Alg.4)	BR	BS	AFK
20	0.098	0.110	0.104	0.112
	(0.09,0.10)	(0.11,0.11)	(0.10,0.11)	(0.11,0.12)
50	0.144	0.148	0.146	0.146
	(0.13,0.16)	(0.14,0.16)	(0.13,0.16)	(0.14,0.15)
100	0.182	0.216	0.194	0.214
	(0.18,0.19)	(0.20,0.23)	(0.19,0.20)	(0.20,0.23)
500	0.698	1.116	0.810	0.986
	(0.63,0.74)	(1.07,1.17)	(0.63,1.00)	(0.85,1.11)
1000	1.390	2.420	1.662	2.180
	(1.27,1.53)	(2.16,2.76)	(1.37,2.01)	(2.01,2.36)
2000	2.812	5.884	2.932	4.800
	(2.34,2.99)	(4.98,6.63)	(2.81,3.18)	(4.30,5.09)
5000	7.456	25.038	7.520	20.172
	(6.82,9.03)	(22.33,27.54)	(6.16,10.12()	(16.57,22.33)
10000	14.330	80.008	15.434	59.634
	(12.45,16.61)	(73.16,87.03)	(12.88,18.12)	(55.60,65.80)

See table 7, for abbreviations.

In the case of four parameters model as shown by table 9, though BS algorithm is competing with algorithm 4, this ordering remains unchanged and algorithm 4 is again most efficient. The ranking of the selected algorithms are similar to that of three parameters experiments in all cases of small, medium and larger sample sizes.

Table 9. CPU times for multiple model (m=4) selected algorithms

n	B(Alg.4)	BR	BS	AFK
20	0.112	0.116	0.116	0.116
	(0.11,0.12)	(0.11,0.12)	(0.11,0.12)	(0.11,0.12)
50	0.156	0.168	0.160	0.160
	(0.15,0.16)	(0.16,0.19)	(0.15,0.17)	(0.16,0.16)
100	0.284	0.286	0.286	0.286
	(0.26,0.32)	(0.26,0.30)	(0.27,0.30)	(0.26,0.30)
500	1.098	1.596	1.260	1.394
	(0.85,1.44)	(1.23,1.77)	(0.92,1.58)	(1.17,1.71)
1000	2.194	4.016	2.200	3.022
	(2.03,2.45)	(3.35,5.05)	(0.64,3.28)	(2.73,3.21)
2000	4.650	10.636	5.430	7.774
	(3.92,5.05)	(9.44,11.86)	(4.54,6.15)	(7.25,8.16)
5000	12.852	41.282	12.938	32.110
	(10.00,15.23)	(32.40,47.86)	(11.62,14.04)	(29.55,33.17)
10000	27.720	119.152	27.864	101.472
	(22.93,39.14)	(107.14,129.70)	(23.94,32.10)	(100.31,103.15)

See table 7, for abbreviations.

When number of parameters increased to five, BS algorithm failed to produce correct answers for sample sizes of 2000 and more. Gentle and Narula and Sposito (1987) also referred to failure of BS algorithm for sample sizes of 1000 and greater for five and more parameters models and for sample size of 5000 when the number of parameters are two. With reference to table 10, efficiency of algorithm 4 to others with respect to failure of BS is clear. The algorithms of AFK and BR are in the next positions respectively. For smaller sample size, BR, BS and AFK algorithms are competing, but the differences are very small. In larger sample size, algorithm 4 becomes strictly superior to other algorithms.

Table 10. CPU times for multiple model (m=5) selected algorithms

n	B(Alg.4)	BR	BS	AFK
20	0.138	0.124	0.124	0.124
	(0.13,0.15)	(0.12,0.13)	(0.12,0.14)	(0.11,0.13)
50	0.208	0.240	0.204	0.188
	(0.18,0.24)	(0.22,0.26)	(0.20,0.21)	(0.18,0.20)
100	0.348	0.380	0.404	0.338
	(0.33,0.37)	(0.34,0.42)	(0.36,0.46)	(0.32,0.36)
500	2.024	2.498	1.754	1.684
	(1.57,2.40)	(2.29,2.68)	(1.34,2.01)	(1.57,1.78)
1000	3.702	5.684	3.364	3.876
	(3.19,4.45)	(4.86,6.28)	(2.93,3.71)	(3.67,4.13)
2000	8.770	15.500	+	9.120
	(7.51,9.41)	(13.04,16.58)	+	(7.91,9.93)
5000	24.418	66.394	+	36.600
	(20.15,27.96)	(59.93,71.90)	+	(33.95,38.67)
10000	53.924	244.072	+	108.406
	(38.35,64.90)	(217.81,270.06)	+	(99.65,119.75)

+ Failed to compute correct answers.
See table 7, for abbreviations.

Table 11. CPU times for multiple model (m=7) selected algorithms

n	B(Alg.4)	BR	BS	AFK
20	0.160	0.164	0.156	0.160
	(0.16,0.16)	(0.16,0.17)	(0.15,0.16)	(0.15,0.17)
50	0.346	0.302	0.328	0.282
	(0.29,0.37)	(0.29.0.31)	(0.31,0.34)	(0.26,0.30)
100	0.706	0.572	0.540	0.530
	(0.58,0.81)	(0.52,0.65)	(0.48,0.62)	(0.47,0.62)
500	3.898	4.486	3.202	2.990
	(3.25,4.54)	(4.03,4.85)	(2.41,3.87)	(2.45,3.48)
1000	9.178	11.448	+	6.568
	(7.03,10.44)	(9.88,12.71)	+	(5.87,7.68)
2000	19.984	31.872	+	14.908
	(18.06,21.55)	(24.52,35.22)	+	(13.56,16.23)
5000	57.286	141.234	+	58.314
	(49.53,64.24)	(129.20,154.03)	+	(49.12,65.17)
10000	130.79	475.826	+	151.526
	(103.20,183.45)	(421.40,521.39)	+	(144.94,165.62)

+ Failed to compute correct answers.
See table 7, for abbreviations.

In table 11 when number of parameters is seven, BS algorithm failed to compute correct answer for sample of sizes 1000 and more. AFK is the best for smaller samples, but for large samples, algorithm 4 is again superior. BR algorithm is in the third position.

In table 12, with ten parameters, BS and AFK algorithms failed to compute correct answers for larger sample size. BR algorithm is the most efficient with respect to accuracy. Algorithm 4 remains in the second position of both computing time and accuracy, except for sample size of 10000, where algorithm 4 is the most efficient.

Table 12. CPU times for multiple model (m=10) selected algorithms

n	B(Alg.4)	BR	BS	AFK
20	0.276	0.212	0.218	0.212
	(0.27,0.29)	(0.20,0.23)	(0.21,0.22)	(0.20,0.23)
50	0.956	0.502	0.492	0.398
	(0.69,1.72)	(0.47,0.54)	(0.43,0.54)	(0.35,0.44)
100	2.414	1.144	0.998	0.776
	(1.55,4.78)	(1.03,1.25)	(0.86,1.11)	(0.66,0.91)
500	13.980	8.446	5.970	5.210
	(11.76,16.30)	(7.88,9.99)	(4.81,6.92)	(4.43,5.78)
1000	62.624	23.506	+	11.144
	(22.23,193.19)	(20.26,26.93)	+	(9.08,13.14)
2000	109.268	62.756	+	+
	(72.20,278.27)	(59.43,65.66)	+	+
5000	409.438	284.618	+	+
	(154.64,1010.92)	(240.02,322.47)	+	+
10000	679.540	967.794	+	+
	(283.04,1076.04)	(770.79,1064.43)	+	+

+ Failed to compute correct answers.
See table 7, for abbreviations.

5. Conclusions

Since in computational algorithms, accuracy is more important than efficiency, those L_1 norm algorithms should be selected which produce correct solutions and among them, the fastest one should be selected. Algorithm 2 and algorithm of Josvanger and Sposito (1983) both computed correct answers for two parameters linear L_1 norm regression model. Algorithm 2 which is faster than JS introduced for applied works.

For multiple regression, BS and AFK algorithms failed to compute correct answers in larger models. As stated by Gentle and Narula and Sposito (1987), because of the accumulated roundoff error, algorithm of Bloomfield and Steiger (1980) was not usable in larger size problems. Coding to avoid rounding problems often increase the execution time, so it is not clear what would happen to the relative efficiency if the BS code is modified. This is also the case for algorithm of Armstrong and Frome and Kung (1979), though it is less sensitive to rounding error than BS algorithm. However, from the previous tables it may be concluded that algorithm 4 is more appropriate for models with less than ten parameters and algorithm of Barrodale and Roberts (1973,74) for the ten parameters model. This last conclusion is not very constructive, because in the case of ten parameters model with 10000 observations algorithm 4 is highly superior to BR. However, since in applied work we are not always confronted with very large amount of data and parameters, this conclusion is poor in operational sense.

References

- N.N. Abdelmalek (1980a) L_1 solution of overdetermined systems of linear equations. ACM Trans. Math. Soft., 6, 220-227.
- N.N. Abdelmalek (1980b) A Fortran subroutine for the L_1 solution of overdetermined systems of linear equations. ACM Trans. Math. Soft., 6, 228-30.
- D. Anderson, W.L. Steiger (1982) A comparison of methods for discrete L_1 curve-fitting. Tech. Rep. DCS-TR-96. Dept. of Comp. Sci., Hill center for the Math. Sci. Busch campus, New Brunswick, N.J.
- R.D. Armstrong, E.L. Frome (1976a) A comparison of two algorithms for absolute deviation curve fitting. JASA, 71, 328-330.
- R.D. Armstrong, E.L. Frome, D.S. Kung (1979) A revised simplex algorithm for the absolute deviation curve fitting problem. Commun. Stat. B8, 175-190.

- R.D. Armstrong, D.S. Kung (1978) AS132: Least absolute value estimates for a simple linear regression problem. Appl. Stat., 27, 363-366.
- Barrodale, F.D.K. Roberts (1973) An improved algorithm for discrete L_1 linear approximation. SIAM J. Numer. Anal., 10, 839-848.
- Barrodale, F.D.K. Roberts (1974) Algorithm 478: Solution of an overdetermined system of equations in the L_1 norm. Commun. ACM, 17, 319-320.
- R.H. Bartels, A.R. Conn, J. Sinclair (1978) Minimization technique for piecewise differentiable functions: The L_1 solution to an overdetermined linear system. SIAM J. Numer. Anal. 15, 224-241.
- B. Bidabad (1987a) Least absolute error estimation. Submitted to the First International Conference on Statistical Data Analysis Based on the L_1 norm and Related Methods, Neuchatel, Switzerland.
- B. Bidabad (1987b) Least absolute error estimation, part II. Submitted to the First International Conference on Statistical Data Analysis Based on the L_1 norm and Related Methods, Neuchatel, Switzerland.
- B. Bidabad (1988a) A proposed algorithm for least absolute error estimation. Proceedings of the Third Seminar of Mathematical Analysis. Shiraz University, 24-34, Shiraz, Iran.
- B. Bidabad (1988b) A proposed algorithm for least absolute error estimation, part II. Proceedings of the Third Seminar of Mathematical Analysis, Shiraz University, 35-50, Shiraz, Iran.
- B. Bidabad (1989a) Discrete and continuous L_1 norm regressions, proposition of discrete approximation algorithms and continuous smoothing of concentration surface, Ph.D. thesis, Islamic Azad University, Tehran, Iran.
- B. Bidabad (1989b) Discrete and continuous L_1 norm regressions, proposition of discrete approximation algorithms and continuous smoothing of concentration surface, Ph.D. thesis, Islamic Azad University, Tehran, Iran. Farsi translation.
- P. Bloomfield, W. Steiger (1980) Least absolute deviations curve fitting. SIAM J. Sci. Statist. Comput. 1, 290-301.
- R. Dutter (1977) Numerical solution of robust regression problems, computational aspects, a comparison. J. of Stat. Computation and Simulation, 5, 207-238.
- J.E. Gentle, S.C. Narula, V.A. Sposito (1987) Algorithms for unconstrained L_1 linear regression. In Y. Dodge (ed.) Statistical data analysis based on the L_1 norm and related methods. North-Holland. 83-94.
- J.E. Gentle, V.A. Sposito, S.C. Narula (1988) Algorithms for unconstrained L_1 simple linear regression. CSDA, 6(4), 335-340.
- J. Gilsinn, K. Hoffman, R.H.F. Jackson, E. Leyendecker, P. Saunder, D. Shier (1977) Methodology and analysis for comparing discrete L_1 approximation codes., Commun. Stat., B6, 399-413.
- K.L. Hoffman, D.R. Shier (1980b) A test problem generator for discrete linear L_1 approximation problems. ACM Trans. Math. Soft., 6, 615-617.
- L.A. Josvanger, V.A. Sposito (1983) L_1-norm estimates for the simple regression problem. Comm. Stat. B12, 215-21.
- W.J. Kennedy, J.E. Gentle (1977) Examining rounding error in least absolute values regression computations. Comm. Stat., B6, 415-420.
- W.J Kennedy, J.E. Gentle, V.A. Sposito (1977a) Comparisons of algorithms for L_1 estimation in the linear model. Paper presented at Midwestern Regional Meeting of IMS, Madison, WI. (Available from the second author).
- W.J Kennedy, J.E. Gentle, V.A. Sposito (1977b) A computer oriented method for generating test problems for L_1 regression. Comm. Stat., B6, 21-27.
- J.L. Larson, A.H. Sameh (1980) Algorithms for round of error analysis a relative error approach. Computing 24, 275-297.
- M.J. Mojarrad (1977) The application of comparative Monte Carlo methods to econometrics: an efficient Monte Carlo study of finite sample properties of iterative instrumental variables estimation. Ph.D. Diss., University of Pennsylvania.
- A.N. Sadovski (1974) AS74: L_1-norm fit of a straight line. Appl. Stat. 23, 244-248.
- E.J. Schlossmacher (1973) An iterative technique for absolute deviations curve fitting. JASA 68, 857-865.
- E. Seneta, W.L. Steiger (1984) A new LAD curve-fitting algorithm: slightly overdetermined equation system in L_1. Discrete Applied Math., 7, 79-91.
- G.O. Wesolowsky (1981) A new descent algorithm for least absolute value regression problem. Comm. Stat., B10, 479-491.