Truncated aggregate homotopy algorithm for the least trimmed squares estimation in nonlinear regression

Yu Xiao[1]*, Zhigang Yan[2]

[1]*School of Basic Science, East China Jiaotong University, Nanchang, Jiangxi, 330013, China*

[2]*School of Accounting, Jiangxi University of Finance and Economics, Nanchang, Jiangxi, 330013, China*

Received 1 October 2014, www.cmnt.lv

Abstract

In data regression, an important role is played by the least trimmed squares (LTS) estimate, which is less sensitive to the outliers than some other estimators such as the least squares estimator. However, estimating the LTS in nonlinear regression would be unimaginable expensive. For the data set with size m and outliers $m-\tilde{m}$, it would require $C_m^{\tilde{m}}$ nonlinear least squares regressions. To solve this problem, this paper studies the LTS solution from an optimization point of view, and proposes truncated aggregate homotopy algorithm to the equivalent min-min-sum programming. Numerical tests with comparisons to some other methods show that the new method is efficient.

Keywords: truncated aggregate homotopy algorithm, least trimmed squares estimation, nonlinear regression

1 Introduction

Since the least squares estimator, l_1, l_2 and l_∞ are affected by wild observations in data regression, Rouseeuw proposed a least trimmed squares (LTS) estimator in [1], which is robust with respect to outliers and applied in many applications. Consider the nonlinear regression model:

$$u_i = f(v_i, x) + r_i, \quad i = 1,2,\cdots,m, \tag{1}$$

where u_i represents the dependent variable, $f(v_i, x)$ is a regression function and r_i is the error item. The LTS estimator is defined as:

$$x_i = \arg\min_{x\in R^n} \sum_{i=1}^{i=\tilde{m}} r_{(i)}^2(x), \tag{2}$$

where the squared residuals are ordered from the smallest to the largest

$$r_{(1)}^2 \le r_{(2)}^2 \le \cdots \le r_{(m)}^2.$$

This is equivalent to finding the $\tilde{m}$ -subset with smallest least squares objective function. For the linear LTS regression, several algorithms have been proposed to compute the estimator exactly. The currently fastest exact algorithm is proposed by Agullo based on a branch-and-bound technique that selects the optimal n-subset without exhaustive evaluation. The BAB algorithm is computationally feasible for data sets with $m \le 50$ and $n \le 5$. But for most data sets the exact algorithms would take too long, and the approximate algorithm should be used. The most used one is PROGRESS proposed by Rousseeuw and Leroy in [2] and developed in [3]. Given a data set, the PROGRESS algorithm examines several n-subsets or elemental sets. For each elemental set, calculates its exact fit of the least squares estimator and the fit with the smallest objective function value provide an approximate LTS estimate. For the nonlinear LTS estimator, i.e., $u = f(v, x)$ is nonlinear with respect to x, the exact solution has to find the LS fit for all $\tilde{m}$ -subsets, which would involve an unmanageable computation cost. The PROGRESS algorithm can also provide rough solution by examining several random $\tilde{m}$ -subsets, but the algorithm has to be repeated in a large number of times to obtain approximate solution.

In this paper, we study the LTS solution from an optimization point of view. Firstly, the LTS model is converted to min-min-sum programming equivalently. Then use the aggregate homotopy method (see [12] for details) to solve the nonsmooth problem. To reduce the computation cost in tracing homotopy curve, we adopt a truncated aggregate smoothing technique, which was proposed in [4] and [5] for solving min-max problems. Moreover, we give some new truncating criterions, for adaptively updating the subset to guarantee the local quadratic convergence of the correction process with as few computational costs as possible.

The paper is organized as follows. In Section 2, the truncated aggregate homotopy method for LTS regression and its convergence are presented. In Section 3, the test results are given.

****Corresponding author's*** mail: xiaoyueajtu@163.com

In the following, for $g(x): R^n \to R^q$, denote $Dg(x) = (\nabla g_1(x), \cdots, \nabla g_q(x))^T$, where $\nabla g_j(x) \in R^n$ is the gradient of $g_j(x)$. Finally, the symbol $\#(I)$ denotes the capacity of set I.

2 Least trimmed squares in nonlinear regression

It is obviously that LTS regression (2) is equivalent to

$$\min_{x \in R^n}\{F(x) = \min_{I \in S} \sum_{i \in I} r_{(i)}^2(x)\}, \tag{3}$$

where $S = \{I \subset \mathrm{q} = \{1, 2, \cdots, m\} \mid \#(I) = \tilde{m}\}$. The first order optimal condition is:

Proposition 1 ([1]) Suppose that $r_i(x)$ is continuously differentiable, Equation (3) obtains minimum at x^*, then there exists $\alpha_I \geq 0$, such that:

$$\sum_{I \in S(x^*)} \alpha_I \sum_{i \in I} \nabla r_i^2(x^*) = 0,$$

$$\sum_{I \in S(x^*)} \alpha_I = 1,$$

where $S(x^*) = \{I \in S \mid \sum_{i \in I} r_i^2(x^*) = F(x^*)\}$.

For this nonconvex and nonsmooth problem, an aggregate function, which was proposed by Li in [6], can be used to smooth the objective $F(x)$ such as

$$F(x,t) = -t \ln\left(\sum_{I \in S} \exp\left(\sum_{i \in I} -r_i^2(x)/t\right)\right), \tag{4}$$

where $t > 0$ is a smoothing parameter, and:

$$F(x) - t \ln C_m^{\tilde{m}} \leq F(x,t) \leq F(x).$$

Then, aggregate homotopy method can be adopted to solve this problem.

Theorem 1 ([7], ***Theorem 3***) If $r_i^2(x) \in C^p (p > 2)$ satisfying the following assumption 1: There exist $M > 0$, $\bar{x} \in R^n$, such that for all $\|x - \bar{x}\| \geq M$, it has:

$$\nabla r_i^2(x)^T (x - \bar{x}) > 0,\ i = 1, 2, \cdots, \in m. \tag{5}$$

Choose a small constant $\theta \in (0,1]$, then for almost all $x^0 \in R^n$, aggregate equation:

$$H_{x^0}(x,t) \equiv (1-t)\nabla_x F(x, \theta t) + t(x - x^0) = 0, \tag{6}$$

determines a smooth curve Γ, which starts from $(x^0, 1)$ and approaches to the plane $t = 0$. Moreover, if $(x^*, 0)$ is a limit point of Γ on the hyperplane $t = 0$, then x^* is a KKT point of Equation(3).

The Predictor-Corrector method (see [8] for details) is usually adopted to numerically trace the homotopy path Γ. Moreover, to reduce the computation cost, we proposed a truncated aggregate homotopy method to trace the homotopy path efficiently for nonlinear programming in [5]. Here, we can also use the truncated aggregate technique. For given $\bar{x} \in R^n$, choose a parameter $\varepsilon > 0$, denotes:

$$\mathrm{q}(\bar{x}, \varepsilon) = \{i \mid r_i^2(\bar{x}) - r_{(\tilde{m})}^2(\bar{x}) \leq \varepsilon, i \in \mathrm{q}\}, \tag{7}$$

$$\bar{S} = \{I \subset \mathrm{q}(\bar{x}, \varepsilon) \mid \#(I) = \tilde{m}\}. \tag{8}$$

The truncate aggregate function with respect to $\mathrm{q}(\bar{x}, \varepsilon)$ is defined as

$$F^{\bar{S}}(x,t) = -t \ln\left(\sum_{I \in \bar{S}} \exp\left(\sum_{i \in I} -r_i^2(x)/t\right)\right). \tag{9}$$

For conciseness, we write $\mathrm{q}_\varepsilon = \mathrm{q}(\bar{x}, \varepsilon)$.

At first, we give some estimates of difference between aggregate function Equation (4) and truncated aggregate function Equation (9), which are important for the efficient implementation of our algorithm. Denote $A_o(x) = \max_{i \in \mathrm{q}} \|r_i^2(x)\|$, $A_1(x) = \max_{i \in \mathrm{q}} \|\nabla r_i^2(x)\|$, $A_2(x) = \max_{i \in \mathrm{q}} \|\nabla^2 r_i^2(x)\|$.

Corollary 1 Suppose that $r_i(x)$, $i \in \mathrm{q}$, are continuously differentiable. For any given $\bar{x} \in R^n$, $0 < t < 1$, $\varepsilon > 0$, let $F(x,t)$, q_ε and $F^{\bar{S}}(x,t)$ be defined as in Equations (4), (7) and (9). Then the following error estimates hold:

(i) $0 \leq F^{\bar{S}}(\bar{x},t) - F(\bar{x},t) \leq t(C_m^{\tilde{m}} - 1)\exp(-\varepsilon/t)$;

(ii) $\|\nabla_x F^{\bar{S}}(\bar{x},t) - \nabla_x F(\bar{x},t)\| \leq \dfrac{2\tilde{m}A_1(\bar{x})(C_m^{\tilde{m}} - 1)}{\exp(\varepsilon/t) + C_m^{\tilde{m}} - 1}$;

(iii) $\|\nabla_x^2 F^{\bar{S}}(\bar{x},t) - \nabla_x^2 F(\bar{x},t)\| \leq \dfrac{2\tilde{m}\left(A_2(\bar{x}) + 3\tilde{m}\dfrac{A_1^2(\bar{x})}{t}\right)(C_m^{\tilde{m}} - 1)}{\exp(\varepsilon/t) + C_m^{\tilde{m}} - 1}$;

(iv) $\|\nabla_t F^{\bar{S}}(\bar{x},t) - \nabla_t F(\bar{x},t)\| \leq \dfrac{\left(1 + 2\tilde{m}\dfrac{A_0(\bar{x})}{t}\right)(C_m^{\tilde{m}} - 1)}{\exp(\varepsilon/t) + C_m^{\tilde{m}} - 1}$;

(v) $\|\nabla_{xt} F^{\bar{S}}(\bar{x},t) - \nabla_{xt} F(\bar{x},t)\| \leq \dfrac{6(C_m^{\tilde{m}} - 1)\tilde{m}^2 A_0(\bar{x}) A_1(\bar{x})}{t^2(\exp(\varepsilon/t) + C_m^{\tilde{m}} - 1)}$.

Proof Here we give the proof for items (i)-(ii), and others can be proofed similarly or refer to the Corollary 3.1 in [5]. For conciseness, denote

$G(\bar{x},t,I)=\exp((F(\bar{x})+\sum_{i\in I}-r_i^2(\bar{x}))/t).$

Then:

$$F^{\bar{S}}(\bar{x},t)-F(\bar{x},t)=t\ln(1+\sum_{I\notin S}G(\bar{x},t,I)/\sum_{\in S}G(\bar{x},t,I))\geq 0.$$

Since there exists at least one $I\in\bar{S}$ satisfying $F(\bar{x})+\sum_{i\in I}-r_i^2(\bar{x})=0$, hence:

$$\sum_{I\in S}G(\bar{x},t,I)\geq 1\,. \tag{10}$$

By the definition of $\bar{S}$ in Equations (7) and (8), for all $I\notin\bar{S}$, it has:

$\sum_{i\in I}r_i^2(\bar{x})\geq \min_{i\notin q_\varepsilon}r_{(i)}^2(\bar{x})+\sum_{i=1}^{\tilde{m}-1}r_{(i)}^2(\bar{x})$, and hence:

$$F(\bar{x})-\sum_{i\in I}r_i^2(\bar{x})=\sum_{i=1}^{\tilde{m}}r_{(i)}^2(\bar{x})-\sum_{i\in I}r_i^2(\bar{x})\leq$$
$$r_{(\tilde{m})}^2(\bar{x})-\min_{i\notin q_\varepsilon}r_i^2(\bar{x})\leq-\varepsilon\,.$$

Then:

$$\sum_{I\notin S}G(\bar{x},t,I)\leq(C_m^{\tilde{m}}-1)\exp(-\varepsilon/t)\,. \tag{11}$$

Substituting equations (10) and (11), together with $\ln(1+x)<x(x>0)$, we obtain:

$$F^{\bar{S}}(\bar{x},t)-F(\bar{x},t)\leq t\sum_{I\notin S}G(\bar{x},t,I)\leq t(C_m^{\tilde{m}}-1)\exp(-\varepsilon/t)\,.$$

$$\begin{aligned}&\|\nabla_xF^{\bar{S}}(x,t)-\nabla_xF(x,t)\|=\\&\|\sum_{I\in\bar{S}}\varsigma_I^{\bar{S}}(x,t)\sum_{i\in I}\nabla r_i^2(x)-\sum_{I\in S}\varsigma_I(x,t)\sum_{i\in I}\nabla r_i^2(x)\|=\\&\|\sum_{I\in\bar{S}}(\varsigma_I^{\bar{S}}(x,t)-\varsigma_I(x,t))\sum_{i\in I}\nabla r_i^2(x)-\sum_{I\notin\bar{S}}\varsigma_I(x,t)\sum_{i\in I}\nabla r_i^2(x)\|\end{aligned}\,, \tag{12}$$

where:

$$\varsigma_I(x,t)=\frac{\exp\left(\sum_{i\in I}-r_i^2(x)/t\right)}{\sum_{I\in S}\exp\left(\sum_{i\in I}-r_i^2(x)/t\right)},I\in S\,;$$

$$\varsigma_I^{\bar{S}}(x,t)=\frac{\exp\left(\sum_{i\in I}-r_i^2(x)/t\right)}{\sum_{I\in\bar{S}}\exp\left(\sum_{i\in I}-r_i^2(x)/t\right)},I\in\bar{S}\,.$$

We first estimate the upper bound of:

$\sum_{I\in\bar{S}}|\varsigma_I^{\bar{S}}(\bar{x},t)-\varsigma_I(\bar{x},t)|$ and $\sum_{I\notin\bar{S}}\varsigma_I(\bar{x},t)$. For all $I\in S$, $\sum_{I\in\bar{S}}\varsigma_I^{\bar{S}}(\bar{x},t)=\sum_{I\in S}\varsigma_I(\bar{x},t)=1, 0\leq\varsigma_I(\bar{x},t)\leq\varsigma_I^{\bar{S}}(\bar{x},t)$, hence:

$$\sum_{I\in\bar{S}}|\varsigma_I^{\bar{S}}(\bar{x},t)-\varsigma_I(\bar{x},t)|=\sum_{I\in\bar{S}}\varsigma_I^{\bar{S}}(\bar{x},t)-\varsigma_I(\bar{x},t)=\sum_{I\notin\bar{S}}\varsigma_I(\bar{x},t).$$

By the definition of $\bar{S}$, it has:

$$\sum_{I\notin\bar{S}}\varsigma_I(\bar{x},t)=\frac{\sum_{I\notin\bar{S}}G(\bar{x},t,I)}{\sum_{I\in\bar{S}}G(\bar{x},t,I)}=\left(1+\frac{\sum_{I\in\bar{S}}G(\bar{x},t,I)}{\sum_{I\notin\bar{S}}G(\bar{x},t,I)}\right)^{-1}.$$

From Equation (11), it obtains:

$$\sum_{I\in\bar{S}}|\varsigma_I^{\bar{S}}(\bar{x},t)-\varsigma_I(\bar{x},t)|=\sum_{I\notin\bar{S}}\varsigma_I(\bar{x},t)\leq$$
$$\frac{(C_m^{\tilde{m}}-1)}{\exp(\varepsilon/t)+C_m^{\tilde{m}}-1}. \tag{13}$$

Then together with Equations (12) and (13), we have:

$$\|\nabla_xF^{\bar{S}}(\bar{x},t)-\nabla_xF(\bar{x},t)\|\leq$$
$$\tilde{m}A_1(\bar{x})\left(\sum_{I\in\bar{S}}|\varsigma_I^{\bar{S}}(x,t)-\varsigma_I(x,t)|+\sum_{I\notin\bar{S}}\varsigma_I(x,t)\right)\leq$$
$$\frac{2\tilde{m}A_1(\bar{x})(C_m^{\tilde{m}}-1)}{\exp(\varepsilon/t)+C_m^{\tilde{m}}-1}.$$

Now, we consider to use the truncated aggregate technique to trace the homotopy path Γ efficiently. For conciseness, we write $v=v(x,t)$.

Algorithm 1 (Truncated aggregate homotopy algorithm)

Data. $\theta>0, x^0\in R^n, t_0=1$.

Parameters. Initial steplength $h_0>0$; tolerance $t_{tol}=10^{-7}$, $t_c=10^{-6}$, $H_{tol}=\min\{10^{-3},1/t\}$; maximum steplength $h_{\max}$; maximum inner iteration number N_{in}; A_1, A_1 are big numbers; errors parameters $\{\eta_{k,i}\}_{k,i=0}^{\infty},\{\mu_{k,i}\}_{k,i=0}^{\infty}$.

Step 0. Unit tangent vector $d^0, k=0, i=0$.

(Predictor step)

Step 1. If $0\leq t_k\leq t_{tol}$, end the procedure; else go to Step 2.

Step 2. If $k>0$, compute $d_k=\dfrac{v^k-v^{k-1}}{\|v^k-v^{k-1}\|}$. If $t_k>t_c$, go to Step 3, else go to Step 4.

Step 3. Set $v^{k,0}=v^k+h_kd^k, i=0,$ go to Step 5.

Step 4. Let $h_k=\dfrac{t_{tol}-t_k}{d_{n+1}^k}$, and $v^{k,0}=v^k+h_kd^k$, then correct $v^{k,0}$ on the hyperplane $t=t_{tol}$.

(Corrector step).

Step 5. If $t_{k,i} \notin [0,1]$, $h_k = 0.5h_k$, set, go to Step 2; else, let $v = (x,t) = (x^{k,i}, t_{k,i}), \eta = \eta_{k,i}, \mu = \mu_{k,i}$, calculate:

$$\varepsilon = \varepsilon^{k,i} = \max\{\bar{\varepsilon}, \bar{\bar{\varepsilon}}\}, \tag{14}$$

where:

$$\bar{\varepsilon} = \theta t \ln(\max\{\frac{(C_m^{\tilde{m}} - 1)(2\tilde{m}A_1 - \eta)}{\eta}, 1\}),$$

$$\bar{\bar{\varepsilon}} = \theta t \ln(\max\{\frac{2\tilde{m}(C_m^{\tilde{m}} - 1)}{\mu}((1-t)\times \tag{15}$$

$$(A_2 + \frac{3\tilde{m}A_1^2}{\theta t} + \frac{3\tilde{m}A_0(x)A_1}{\theta t^2}) + A_1) - (C_m^{\tilde{m}} - 1), 1\}),$$

go to Step 6.

Step 6. If $\| R_{k,\varepsilon}(v) \| \le H_{tol}$, where

$$R_{k,\varepsilon}(v) = \begin{bmatrix} H_\varepsilon(v) \\ d^{k^T}(v - v^{k,0}) \end{bmatrix},$$

with $H_\varepsilon(v) = (1-t)\nabla_x F^{\bar{S}}(x, \theta t) + t(x - x^0)$, go to Step 7; else go to Step 8.

Step 7. Set $v^{k+1} = v^{k,i}, h_{k+1} = \begin{cases} \min\{1.5h_k, h_{\max}\}, i < 3, \\ h_k, \quad \quad else, \end{cases}$

$k = k+1$, go to Step 1.

Step 8. If $i \ge N_{in}$, set $h_k = 0.5h_k$, go to Step 3; else, obtain $v^{k,i+1}$ using truncated aggregate iteration, $v^{k,i+1} = v^{k,i} - DR_{k,\varepsilon}(v^{k,i})^{-1} R_{k,\varepsilon}(v^{k,i})$, set $i = i+1$, and go to Step 5.

Then, from Corollary 1, the following proposition holds.

Proposition 2 Suppose that $r_i^2(x), i \in q$ are twice continuously differentiable. In Algorithm 1, for any iteration point v, and any error parameters $\eta, \mu > 0$, if ε is set as Equations (14) and (15) with $A_1 \ge A_1(x)$ and $A_2 \ge A_2(x)$, then it has:

$$\| H(v) - H_\epsilon(v) \| \le \eta, \tag{17}$$

$$\| DH(v) - DH_\epsilon(v) \| \le \mu. \tag{18}$$

Proof: Since:

$$\epsilon \ge \bar{\epsilon} = \theta t \ln\left(\max\left\{ \frac{(C_m^{\tilde{m}} - 1)(2(1-t)\tilde{m}A_1 - \eta)}{\eta}), 1 \right\} \right),$$

the following inequality holds:

$$2(1-t)\tilde{m}A_1(C_m^{\tilde{m}-1)} / (\exp(\epsilon / (\theta t)) + C_m^{\tilde{m}} - 1) \le \eta.$$

According to Corollary 1, we have:

$$\left\| H(v) - H_\epsilon(v) \right\| = (1-t)\left\| \nabla_x F(x, \theta t) - \nabla_x F^{\bar{S}}(x, \theta t) \right\| \le$$

$$(1-t)2\tilde{m}A_1(x)(C_m^{\tilde{m}-1)} / (\exp(\epsilon / (\theta t)) + C_m^{\tilde{m}} - 1),$$

Since $A_1 \ge A_1(x)$, it immediately follows from Equations (20) and (21) that $\| H(v) - H_\epsilon(v) \| \le \eta.$

Similarly, the assertion $\| DH(v) - DH_\epsilon(v) \| \le \mu$ also holds.

The global convergence proofs for PC algorithms have been given in [8-11] and some other papers. Moreover, under some proper assumptions, the local quadratic convergence of corrector step with truncated aggregate technique was also discussed in [5]. Here, we only give the convergence results, and the proofs can refer to [5].

Theorem 2 Suppose that $r_i^2(x) \in C^p(p > 2)$ satisfying assumption 1. In Algorithm 1, assume $v^k \in H^{-1}(0)$, and $\begin{bmatrix} DH(v^k) \\ d^{kT} \end{bmatrix}$ is non-singular. Then there exist $h > 0, \eta_{k,i} > 0, \mu_{k,i} > 0$, such that:

(i) The sequence $\{v^{k,i}\}$ generated by the corrector steps in Algorithm 1 is well defined and finite;

(ii) For sufficiently small H_{tol} and $v^{k,i}$ sufficiently close to v^{k+1}, *it has* $\| v^{k,i+1} - v^{k+1} \| = O(\| v^{k,i} - v^{k+1} \|^2)$.

3 Numerical experiment

In this section, we give some numerical results, comparing Algorithm 1 (TAH) with some other algorithms, such as the PROGRESS algorithm and aggregate homotopy method (AH) in [12], to show the efficiency of our algorithm.

During the computation, we set parameters $h_0 = 0.1$, $h_{max} = 0.3, \theta = 0.1$ or 0.01, $\eta_{k,i} = 1e-1, \mu_{k,i} = 1e2$ for all $k, i \in N, A_1 = A_2 = 1e2$. The parameters p in PROGRESS algorithm is set as 10000. All the computations are done by running MATLAB 7.6.0 on a laptop with AMD Turion (tm) 64 $\times$ 2 Mobile Technology TL-58 CPU 1.9 GHz and 896M memory.

The numerical results are reported in the following tables, in which x^* denotes the final approximate solution, F^* is the value of the objective function at x^*, time is the CPU time in seconds.

Example 1 Rotated cone fitting [13]. We test our algorithm for the artificial rotated cone data points which are generated as that in [13]. At first, produce 30 points on an unrotated cone with error item, and then make rotation and translation to these data (see [14,15] for details). The final data with 6 outliers are listed in Table A1.

Example 2 Hyperspheres fitting [16]. In this example, we try to fit a hypersphere to a set of points on its surface.

The data listed in Table A2 are distributed on a hypersphere in 8 dimensional space, in which 32 points are with slight perturbing item following Gaussian distribution and 8 points with strong perturbing. Denote the centre and radius of hypersphere as $o \in R^8$ and $l \in R$, respectively. Then the error item of u_i is $r_i^2 = (\| u_i - o \| - l)^2$.

TABLE 1 The numerical results of example 1, $x^0 = (-0.1,-0.1,-2,1.5,-1.5,0.8)^T$

Method	x^*	F^*	time
TAH	(–0.184274, –0.125505, –2.100411,1 .318975, –1.307862,0 .521104)	0.040820	2.004
AH	(–0.184274, –0.125505, –2.100411,1 .318975, –1.307862, 0.521104)	0.040820	12.501
PROGRESS	(0.052362, –0.002260, –2.475498, 2.078406, –1.318664, 0.525396)	0.278631	221.860

TABLE 2 The numerical results of example 2, $o^0 = (-1.2,1.2,-1.2,1.2,-1.2,1.2,-1.2,1.2)^T, l^0 = 1.$

Method	o^*	l^*	F^*	time
TAH	$(-1.239475, 1.207438, -1.161166, \cdots, 1.291455)$	1.018385	0.096960	68.650
AH	$(-1.239475, 1.207438, -1.161166, \cdots, 1.291455)$	1.018385	0.096960	475.250
PROGRESS	$(-1.339223, 1.269973, -1.219799, \cdots, 1.168995)$	1.094665	0.234161	414.160

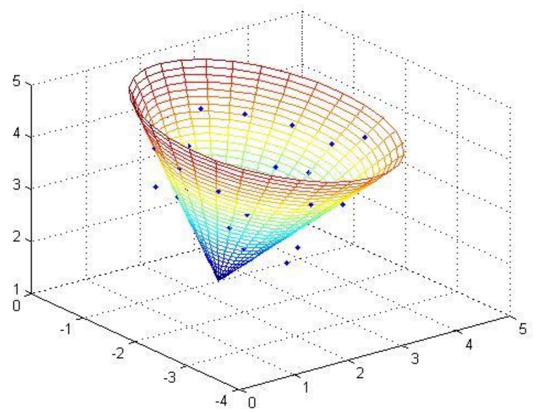

FIGURE 1 Result of example 1.

Results in Tables 1 and 2 show the efficiency of the truncated aggregate homotopy algorithm to the LTS estimator. Compared with some other algorithms, it obtains better solution in lower computation cost. And it can be seen from the Figure 1 that the LTS estimator solved by truncated aggregate homotopy algorithm is robust with respect to outliers. Moreover, it has been shown in [5] that the performance of truncated aggregate homotopy algorithm moderately depends on the values of parameters $A_1, A_2, \eta_{k,i}$ and $\mu_{k,i}$.

5 Conclusions

LTS is a robust estimator, used to be solved approximatively in random algorithms such as PROGRESS. This paper studies the LTS solution from an optimization point of view and proposes a truncated aggregate hmotopy algorithm which has been proved to be more efficient than PROGRESS algorithm.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (11126172 and 11261019), and Science and Technology Project Founded by the Education Department of Jiangxi Province (GJJ14380).

Appendix Data for examples

TABLE A1 Data for Example 1

u_1	(2.806534,–1.554287,2.274129)	u_{11}	(2.350218,–0.569362,3.436930)	u_{21}	(1.525866,–0.697709,4.079642)
u_2	(2.021418,–1.169156,3.038737)	u_{12}	(1.481760,–0.847094,3.028508)	u_{22}	(1.124833,–1.197036,4.021855)
u_3	(1.941398,–2.057755,2.256676)	u_{13}	(1.213677,–1.743698,3.470024)	u_{23}	(0.985255,–1.821152,3.959047)
u_4	(2.542807,–0.800807,2.791917)	u_{14}	(1.541735,–2.534250, 3.241958)	u_{24}	(1.116588,–2.423919,3.761449)
u_5	(1.594323,–1.145537,2.947376)	u_{15}	(2.278532,–2.761561, 2.515192)	u_{25}	(1.606764,–3.027052,4.359070)
u_6	(1.597192,–2.154629,2.810543)	u_{16}	(3.170966,–2.513519, 3.111065)	u_{26}	(2.097407,–3.187533,3.590662)
u_7	(2.471767,–2.339835,1.968811)	u_{17}	(3.708687,–1.150362, 3.727169)	u_{27}	(2.719968,–3.156690,3.395929)
u_8	(3.140913,–1.608041,2.613522)	u_{18}	(3.307613,–0.657566, 3.826187)	u_{28}	(3.309112,–2.888588,3.484840)
u_9	(3.495099,–1.686322,3.107765)	u_{19}	(2.736191,–0.397055,3.963031)	u_{29}	(3.778498,–2.473904,4.077043)
u_{10}	(3.154119,–0.879770,3.234812)	u_{20}	(2.091500,–0.400494,3.984749)	u_{30}	(3.849329,–1.764658,3.602360)

TABLE A2 Data for Example 2

u_1	$(-1.827137,1.620255,-1.360230,1.616242,-1.000295,1.591458,-1.036282,0.351280)$
u_2	$(-1.722459,2.222494,-1.115138,1.483823,-1.183238,0.791812,-1.616898,1.51660)$
u_3	$(-1.653892,1.567384,-1.029011,1.603645,-0.785896,1.626411,-0.854843,1.235471)$
u_4	$(-0.987947,1.387279,-1.431328,1.220211,-1.434079,1.074043,-1.562894,1.846087)$
u_5	$(-1.290262,1.816108,-0.538461,0.675823,-1.315556,1.196824,-1.360835,1.174805)$
u_6	$(-1.327590,0.685783,-0.413155,0.927266,-1.022113,1.284149,-1.101858,1.307970)$
u_7	$(-1.165515,1.323575,-1.246272,1.714151,-1.311673,2.048497,-1.155149,1.182361)$
u_8	$(-0.831673,1.217852,-1.416134,1.167204,-1.011977,1.927850,-0.975847,1.839260)$
u_9	$(-1.418247,1.418085,-0.432122,1.631491,-0.971979,1.702807,-1.388405,1.075423)$
u_{10}	$(-1.716231,1.635902,-0.606673,0.712868,-1.275228,1.101567,-1.023334,1.497552)$
u_{11}	$(-1.328284,1.391991,-1.278199,1.373466,-0.955303,0.714438,-1.150693,1.495779)$
u_{12}	$(-0.628547,1.235044,-1.391275,1.579511,-1.277267,1.011932,-0.977804,0.589740)$
u_{13}	$(-0.541323,1.601350,-1.182552,0.972803,-1.835127,1.144766,-1.361458,1.478742)$
u_{14}	$(-0.456780,1.355549,-1.445953,1.900065,-1.292600,1.875227,-0.865732,0.718817)$
u_{15}	$(-0.614772,1.468949,-1.028108,0.985335,-1.281291,1.065213,-1.698331,1.326452)$
u_{16}	$(-1.102485,1.458414,-1.718008,1.865495,-1.124556,1.377472,-1.392416,1.129426)$
u_{17}	$(-1.378289,0.977239,-0.904086,1.190805,-0.952493,1.314183,-1.189239,1.320107)$
u_{18}	$(-1.142382,0.690286,-1.053657,1.217858,-1.027151,1.490543,-1.262483,1.061356)$
u_{19}	$(-1.640349,1.044617,-1.762241,1.056616,-1.288195,0.841929,-1.219371,1.862674)$
u_{20}	$(-0.956155,0.957255,-0.588287,1.491942,-1.203563,1.352373,-0.795164,0.845923)$
u_{21}	$(-0.579823,2.053821,-0.706565,1.126505,-1.774608,1.316290,-1.083621,1.055077)$
u_{22}	$(-0.983528,0.528166,-0.503125,1.406587,-1.423392,1.220692,-0.873650,1.394656)$
u_{23}	$(-1.081432,0.841913,-0.907088,0.948535,-1.347757,1.325782,-1.097644,1.132886)$
u_{24}	$(-0.709620,0.991185,-1.544125,1.765878,-0.735498,1.406039,-1.059873,1.491111)$
u_{25}	$(-1.164395,0.952351,-0.740086,0.738891,-0.856732,1.590017,-1.486094,1.058320)$
u_{26}	$(-1.443198,1.355000,-1.522945,0.863073,-1.791963,1.060749,-1.301074,0.904337)$
u_{27}	$(-1.740240,0.857176,-1.521763,1.370470,-1.568739,0.574294,-1.450020,1.485106)$
u_{28}	$(-1.247534,1.027606,-0.960262,1.659943,-0.794341,1.940597,-1.031388,1.294884)$
u_{29}	$(-1.255142,1.706268,-0.672697,1.570197,-1.474390,0.657427,-0.727615,1.059744)$
u_{30}	$(-1.453926,1.092562,-1.205667,0.784380,-1.348050,1.456884,-1.840472,1.283986)$
u_{31}	$(-1.616538,1.828746,-0.806913,1.062878,-0.996941,0.819289,-1.281743,1.541216)$
u_{32}	$(-1.387191,1.291204,-1.549861,1.480548,-1.092978,1.417954,-1.639727,0.584830)$
u_{33}	$(-1.638255,0.605401,-0.670375,0.922830,-1.295767,0.760855,-1.063272,0.912512)$
u_{34}	$(-1.264501,1.587916,-1.149242,1.626389,-1.133297,1.460121,-1.841580,1.220608)$
u_{35}	$(-1.119004,0.937540,-0.881953,1.863364,-1.382071,0.908378,-0.796257,0.939305)$
u_{36}	$(-0.947734,1.226757,-0.860558,1.369251,-1.395788,1.253462,-0.843488,0.451297)$
u_{37}	$(-1.260089,1.267644,-1.184468,0.528786,-0.968197,1.853219,-0.959344,1.656337)$
u_{38}	$(-1.820868,1.168449,-0.849195,1.484691,-1.451238,0.573802,-0.993819,1.056483)$
u_{39}	$(-1.361233,1.196598,-1.566313,1.222775,-1.117747,1.792952,-1.695608,1.459923)$
u_{40}	$(-0.962280,1.024718,-0.814362,0.710661,-1.352855,0.784932,-1.363394,0.757177)$

References

[1] Clarke F H 1983 Optimization and Nonsmooth Analysis *John Wiley & Sons Inc.* New York

[2] Rousseeuw P J, Leroy A M 1987 Leroy, Robust Regression and Outlier Detection *John Wiley* New York

[3] Rousseeuw P J, Hubert M 1997 *Recent developments in PROGRESS. In L_1- Statistical Procedures and Related Topics The IMS Lecture Notes – Monograph Series* **31** Institute of Mathematical Statistics Hayward CA 201-14

[4] Xiao Y, Xiong H J, Yu B 2010 Truncated aggregate smoothing Newton method for minimax problems *Appl Math* **216** 1868-79

[5] Xiao Y, Xiong H J, Yu B 2014 Truncated aggregate homotopy method for nonconvex nonlinear programming *Mathematics Methods and Software* **29**(1) 160-176.

[6] Li X S 1991 An aggregate function method for nonlinear programming *Sci China Ser* A **34**(2) 1467-73 *(in Chinese)*

[7] Liu G X 2003 Aggregate homotopy methods for solving sequential max-min problems, complementarity problems and variational inequalities *PhD thesis* Jilin university: China *(in Chinese)*

[8] Allgower E L, Georg K 1997 Numerical Path Following *Handbook of Numerical Analysis* **5** North-Holland 3-207

[9] Keller H B 1977 Numerical solution of bifurcation and nonlinear eigenvalue problems, in Application of Bifurcation Theory *H Academinc Press* New York 359-84

[10] Menzel R, Schwetlick H 1978 Zur Losung parameter abhangiger nicht linearer Gleichungen mit singularen Jacobi-Matrizen *Numerische Mathematik* **30** 65-79

[11] Rheinboldt W C 1980 Solution fields of nonlinear equations and continuation methods *SIAM J Numer Anal* **17**(2) 221-37

[12] Yu B, Feng G C, Zhang S L 2001 The aggregate constraint homotopy method for nonconvex nonlinear programming *Nonlinear Anal TMA* **45**(7) 839-47

[13] Spath H 2001 Least squares fitting with rotated paraboloids *Math Comm* **6** 173-9

[14] Xiao Y, Yu B, Wang D L 2010 Truncated smoothing Newton method for L_∞ fitting rotated cones *Journal of Mathematical Research & Exposition* **30**(1) 159-66

[15] Xiao Y 2014 Truncated aggregate smoothing Newton method for computing LTS estimator *Journal of East China Jiaotong University* **31**(4) 59-64 *(in Chinese)*

[16] Zelniker E E, Clarkson I V L 2004 A generalisation of the Delogne-Kasa method for fitting hyperspheres *Proceedings of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computing* **2** 2069-73

Authors

Yu Xiao, February 1981, Hunan, China.

Current position, grades: Doctor at East China Jiaotong University.
University studies: School of mathematical sciences. Doctor's degree in Dalian University of Technology.
Scientific interests: numerical optimization, data mining, image processing.
Publications: 6 papers.

Zhigang Yan, June 1978, Henan, China.

Current position, grades: Doctor at Jiangxi University of Finance and Economics.
University studies: School of Accounting. Master's degree at Dongbei University of Finance and Economics in 2004. Doctor's degree at Dongbei University of Finance and Economics in 2013.
Scientific interests: financial data mining.
Publications: 7 papers.