Hindawi Publishing Corporation
ISRN Operations Research
Volume 2013, Article ID 393482, 7 pages
http://dx.doi.org/10.1155/2013/393482

Research Article

A Spline Smoothing Newton Method for L_∞ Distance Regression with Bound Constraints

Li Dong[1,2] and Bo Yu[1]

[1] *School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning 116024, China*
[2] *College of Science, Dalian Nationalities University, Dalian, Liaoning 116605, China*

Correspondence should be addressed to Bo Yu; yubo@dlut.edu.cn

Received 15 February 2013; Accepted 19 March 2013

Academic Editors: I. Ahmad and X.-M. Yuan

Orthogonal distance regression is arguably the most common criterion for fitting a model to data with errors in the observations. It is not appropriate to force the distances to be orthogonal, when angular information is available about the measured data points. We consider here a natural generalization of a particular formulation of that problem which involves the replacement of l_2 norm by l_∞ norm. This criterion may be a more appropriate one in the context of accept/reject decisions for manufacture parts. For l_∞ distance regression with bound constraints, we give a smoothing Newton method which uses cubic spline and aggregate function, to smooth max function. The main spline smoothing technique uses a smooth cubic spline instead of max function and only few components in the max function are computed; hence it acts also as an active set technique, so it is more efficient for the problem with large amounts of measured data. Numerical tests in comparison to some other methods show that the new method is very efficient.

1. Introduction

For fitting curves or surfaces to observed or measured data, a common criterion is orthogonal distance regression (ODR). Given the data pairs (u_j, w_j), $j = 1, \dots, m$, where $u_j \in R^{p_1}$ is the independent variable and $w_j \in R^1$ is the dependent variable; suppose

$$w_j = g(d, u_j), \quad (1)$$

where $d \in R^{p_2}$ is a vector of parameters to be determined. We assumed that δ_j is the random error associated with u_j and that ε_j is the random error associated with w_j. To be precise, we relate the quantities u_j, w_j, δ_j, and ε_j to

$$w_j = g(d, u_j + \delta_j) + \varepsilon_j, \quad j = 1, 2, \dots, m. \quad (2)$$

As shown in Boggs et al. [1] this gives rise to the ODR problem given by

$$\min_{d,\delta} \frac{1}{2} \sum_{j=1}^{m} \left[\left(w_j - g(d, u_j + \delta_j) \right)^2 + \delta_j^2 \right]. \quad (3)$$

The ODR problem can be solved by the Gauss-Newton or Levenberg-Marquardt methods (see [1, 2]). The general form of the bounded constrained ODR problem can be expressed by

$$\begin{aligned} &\min_{d,\delta} \quad \frac{1}{2} \sum_{j=1}^{m} \left[\left(w_j - g(d, u_j + \delta_j) \right)^2 + \delta_j^2 \right], \\ &\text{s.t.} \quad L_d \le d \le U_d, \end{aligned} \quad (4)$$

where L_d and U_d are vectors of length p_2 that provide the lower and upper bounds on d, respectively. Zwolak et al. give the algorithm to handle (4) in [3].

It is not appropriate to force the distances to be orthogonal, when angular information is available about the measured data points, such as the rotated cone fitting problem and rotated paraboloid fitting problem. Then, (4) becomes

$$\begin{aligned} &\min_{x,\delta} \quad \frac{1}{2} \sum_{j=1}^{m} \left[\left(\overline{w}_j(x) - g(x, \delta_j) \right)^2 + \delta_j^2 \right], \\ &\text{s.t.} \quad L_x \le x \le U_x, \end{aligned} \quad (5)$$

where $(\overline{u}_j, \overline{w}_j) = ((u_j + a), (w_j + b))T(\mu)$, $T(\mu)$ is an orthogonal matrix, $a \in R^{p_1}$, $b \in R^1$, $\mu \in R^{p_3}$, $x = (a, b, \mu, d)$, and L_x and U_x are vectors of length $p = p_1 + 1 + p_3 + p_2$.

When the least squares norm is not appropriate, problem (5) can be generalized to use other measures in a variety of ways. Most generalizations have been of formulations (5), with the l_2 norms replaced with other norms. We consider here l_∞ norms. It may be a more appropriate one in the context of accept/reject decisions for manufacture parts (see [4]). In this paper, we consider the following l_∞ distance regression with bound constraints.

Let

$$\overline{f}_j(x,\delta) = \begin{cases} \overline{w}_j(x) - g\left(x, \delta_j\right) & j = 1, \dots, m \\ \delta_{j-m} & j = m+1, \dots, 2m, \end{cases} \tag{6}$$

$$\begin{aligned} &\min_{x,\delta} \quad f(x,\delta) = \max_{1 \le j \le 2m} \left|\overline{f}_j(x,\delta)\right|, \\ &\text{s.t.} \quad x \in \Lambda = \left\{x \mid L_x \le x \le U_x\right\}. \end{aligned} \tag{7}$$

We know (7) is a minimax problem. There are several different algorithms that have been taken to solve (7), such as subgradient methods (see [5]), SQP methods (see [6–8]), bundle-type methods (see [9–12]), and smooth approximation methods (see [13–21]). For (7), $|\overline{f}_i(x)|$ is nonsmooth function including the absolute value function. Moreover, when large amounts of measured data are to be fitted to a model, the number of components in the maximum function is very large. It is necessary to develop efficient solution methods for problem (7).

In this paper, we consider to uniformly approximate $\overline{f}(x,\delta)$ by the smooth splines introduced in [22].

Let us first recall the formulation of multivariate splines. Let D be a polyhedral domain of R^m which is partitioned with irreducible algebraic surfaces into cells $\Delta = \{\Delta_i \mid i = 1, \dots, N\}$. A function $s(z)$ defined on D is called a k-spline function with rth order smoothness, expressed for short as $s(z) \in S_k^r(D, \Delta)$, if $s(z) \in C^r(D)$ and $s(z)|_{\Delta_i} = p_i \in P_k$, where P_k is the set of all polynomials of degree k or less in m variables. Similar to the smooth splines which uniformly approximate $\min\{z_1, z_2, \dots, z_m\}$ given in [22], we can construct a spline function $s_3^2(z;\varepsilon) \in S_3^2(R^m, \Delta_{\mathrm{MS}}^2)$ to uniformly approximate $\max\{z_1, z_2, \dots, z_m\}$ (as $\varepsilon \to +0$), where Δ_{MS}^2 is the homogenous Morgan-Scott partition of type two in [22], as follows:

$$\begin{aligned} s_3^2(z_1, z_2, \dots, z_m; \varepsilon) = z_{i_1} + \sum_{l=1}^{k-1} c_l \left(l z_{i_{l+1}} - \sum_{j=1}^{l} z_{i_j} + \varepsilon \right)^3, \\ \text{for } z \in \Delta_{i_1 \cdots i_k}(\varepsilon), \end{aligned} \tag{8}$$

where $c_1 = 1/(6\varepsilon^2)$, $c_k / c_{k+1} = (k+2)/k$, $1 \le k \le m$, and the cell $\Delta_{i_1 \cdots i_k}(\varepsilon)$ is the region defined by the following inequalities:

$$\begin{aligned} z_{i_l} - z_{i_{l+1}} \ge 0, &\quad \text{when } 1 \le l < k, \\ (k-1) z_{i_k} - \sum_{j=1}^{k-1} z_{i_j} + \varepsilon \ge 0, & \\ k z_{i_l} - \sum_{j=1}^{k} z_{i_j} + \varepsilon \le 0, &\quad \text{when } k+1 \le l \le m. \end{aligned} \tag{9}$$

The spline smoothing technique uses a smooth cubic spline instead of max function, and only few components in the max function are computed; hence it acts also as an active set technique, so it is more efficient for the minimax problems with nonsmoothness and large numbers of components.

For that (7) is a minimax problem with bound constraints, and we cannot utilize SSN algorithm in [23] directly to solve (7). Here, we try to extend the idea of SSN algorithm to solve it. At first, we use penalty function to transform (7) into an unconstrained minimax problem. Then, using the smooth approximation, a smoothing Newton method (SN) can be used to solve the l_∞ distance regression with bound constraints.

2. The SN Algorithm for l_∞ Distance Regression with Bound Constraints

Firstly, some deformations for (7) are necessary. Due to $|\overline{f}_j(x,\delta)| = \max\{\overline{f}_j(x,\delta), -\overline{f}_j(x,\delta)\}$, then (7) is equivalent to

$$\begin{aligned} &\min_{x,\delta} \quad f(x,\delta) = \max_{1 \le j \le 4m} \left\{f_j(x,\delta)\right\}, \\ &\text{s.t.} \quad x \in \Lambda, \end{aligned} \tag{10}$$

where

$$f_j(x,\delta) = \begin{cases} \overline{f}_j(x,\delta) & j = 1, \dots, 2m \\ -\overline{f}_{j-2m}(x,\delta) & j = 2m+1, \dots, 4m. \end{cases} \tag{11}$$

Assumption 1. We assume that the functions $f_j(x,\delta)$, $j = 1, \dots, 4m$, are twice continuously differentiable.

Let $\phi(x) = \max\{\phi_1(x), \dots, \phi_{2p}(x)\}$, where $\phi_1(x) = x_1 - U_{x,1}, \dots$, and $\phi_p(x) = x_p - U_{x,p}$, $\phi_{p+1}(x) = -x_1 - L_{x,1}, \dots$, $\phi_{2p}(x) = -x_p - L_{x,p}$. Denote the unknown variables x, δ to be x; then $x \in R^n$, where $n = p + p_1$. Use penalty function with penalty parameter $C > 0$ to transform (10) into the following unconstrained programming problem:

$$\min \left\{\psi(x) = \left\{\max_{1 \le j \le 4m} f_j(x) + C\phi(x)\right\}\right\}. \tag{12}$$

The following proposition concerning Theorem 4.2.8 in [24] gives the first-order optimality condition for (12).

Proposition 2. *Suppose that Assumption 1 holds, and then if* (12) *attains the extremum at* x^*, *then*

$$0 \in \partial\psi(x^*) \triangleq \underset{j\in p(x^*)}{\operatorname{conv}}\left\{\partial f_j(x^*)\right\} + C \underset{i\in q(x^*)}{\operatorname{conv}}\left\{\partial\phi_i(x^*)\right\}, \quad (13)$$

where $s(x^*) = \{j \in \mathbf{s} = \{1,\ldots,4m\} \mid f_j(x^*) = f(x^*)\}$, $q(x^*) = \{i \in \mathbf{q} = \{1,\ldots,2p\} \mid \phi_i(x^*) = \phi(x^*)\}$.

We use the following cubic spline to smooth $f(x)$ and the aggregate function to smooth $\phi(x)$ in (12).

Consider

$$\Psi_t(x) = F_t(x) + C\Phi_t(x), \quad (14)$$

where

$$F_t(x) = s_3^2\left(f_{1,t}(x), f_{2,t}(x), \ldots, f_{4m,t}(x); t\right),$$
$$\Phi_t(x) = t\ln\left(\sum_{i=1}^{2p}\exp\left(\frac{\phi_i(x)}{t}\right)\right). \quad (15)$$

Remark 3. Under Assumption 1, $F_t(x)$ and $\Phi_t(x)$ are twice continuously differentiable for arbitrary $t > 0$.

From Lemma 1.1 in [23], Proposition 3.3 in [25], and Proposition 2.4 in [26], we have the following proposition.

Proposition 4. (1) *For any* $x \in \Lambda$, $F_t(x)$ *and* $\Phi_t(x)$ *are monotonically increasing with respect to* $t > 0$.

(2) *Suppose that Assumption 1 holds. Then, for any* $t > 0$, $\Psi_t(x)$ *is twice continuously differentiable, and*

$$\nabla F_t(x) = \sum_{i=1}^{4m}\lambda_{i,t}(x)\nabla f_{i,t}(x) = \sum_{j=1}^{k}\lambda_t^{i_j}(x)\nabla f_{i_j,t}(x), \quad (16)$$

where

$$\lambda_t^{i_j}(x) = \begin{cases} 1 - 3\sum_{l=1}^{k-1}c_l\left(h_l(x,t)\right)^2 & j = 1 \\ 3(j-1)c_{j-1}\left(h_{j-1}(x,t)\right)^2 & \\ \quad -3\sum_{l=j}^{k-1}c_l\left(h_l(x,t)\right)^2 & 2 \le j < k \\ 3(k-1)c_{k-1}\left(h_{k-1}(x,t)\right)^2 & j = k \\ 0 & k < j \le 4m, \end{cases} \quad (17)$$

and $h_l(x,t) = lf_{i_{l+1},t}(x) - \sum_{r=1}^{l} f_{i_r,t}(x) + t$,

$$\nabla\Phi_t(x) = \sum_{i\in\mathbf{q}}\xi_{i,t}(x)\nabla\phi_i(x), \quad (18)$$

where

$$\xi_{i,t}(x) = \frac{\exp(\phi_i(x)/t)}{\sum_{i\in\mathbf{q}}\exp(\phi_i(x)/t)} \in (0,1], \qquad \sum_{i\in\mathbf{q}}\xi_{i,t}(x) = 1. \quad (19)$$

Lemma 5. *Suppose that Assumption 1 holds, Then, for every bounded set* $S \subset R^n$, *there exists an* $L < \infty$ *such that*

$$\left\langle y, \nabla^2\Psi_t(x)\,y\right\rangle \le \frac{1}{t}L\|y\|^2, \quad (20)$$

for all $x \in S$, $y \in R^n$, *and* $0 < t \le 1$.

Proof. From Proposition 4, Lemma 1.2 in [23], and Lemma 2.2 in [17], we know that for every bounded set $S \subset R^n$, there exist $L_F < \infty$ and $L_\Phi < \infty$ such that $\langle y, \nabla^2 F_t(x)y\rangle \le (1/t)L_F\|y\|^2$ and $\langle y, \nabla^2\Phi_t(x)y\rangle \le (1/t)L_\Phi\|y\|^2$ for all $x \in S$, $y \in R^n$, and $0 < t \le 1$. We also know $\nabla^2\Psi_t(x) = \nabla^2 F_t(x) + C\nabla^2\Phi_t(x)$. Let $L = L_F + CL_\Phi$; then $\langle y, \nabla^2\Psi_t(x)y\rangle \le (1/t)L\|y\|^2$. □

Algorithm 6 (The smoothing Newton algorithm).

Step 0. $x^0 \in \Lambda; t_0 > 0, \hat{t} \gg 1$, and α, β, and $\kappa_1 \in (0,1); \kappa_2 \gg 1$, $0 < \kappa_3 \ll 1$ and $\delta > 0$; functions $\epsilon_a(t)$, $\epsilon_b(t)$ and $\tau(t) : (0,\infty) \to (0,\infty)$, satisfying $\epsilon_b(t) \ge \epsilon_a(t) > \tau(t)$ for all $t > 0$. Set $i = 0$, $k = 0$, $s = 1$, and $x^{k,i} = x^0$.

Search the Cell

Step 1. Let $\bar{I} = \{j \mid \max\{f_{1,t_k}(x^{(k,i)}), \ldots, f_{4m,t_k}(x^{(k,i)})\} - f_{j,t_k}(x^{(k,i)}) < t_k\}$; $\bar{k}$ is the cardinality of $\bar{I}$, and $\bar{I} = \{i_1, i_2, \ldots, i_{\bar{k}}\}$; range $\{f_{i_j,t_k}(x^{(k,i)})\}_{j=1}^{\bar{k}}$ according to $f_{i_1,t_k}(x^{(k,i)}) \ge f_{i_2,t_k}(x^{(k,i)}) \ge \cdots \ge f_{i_{\bar{k}},t_k}(x^{(k,i)})$. If $\bar{k} = 1$, the cell is $\Delta_{i_1}(t_k)$. Else, for every $\tilde{k} \in \{\bar{k}, \bar{k}-1, \ldots, 2\}$, if $(\tilde{k}-1)f_{i_{\tilde{k}},t_k}(x^{(k,i)}) - \sum_{j=1}^{\tilde{k}-1} f_{i_j,t_k}(x^{(k,i)}) + t_k \ge 0$, we have $\tilde{k} \in I \subseteq \{\bar{k}, \bar{k}-1, \ldots, 2\}$. Let $\hat{k}$ be the maximum element of I, then the cell is $\Delta_{i_1\cdots i_{\hat{k}}}(t_k)$.

The Stabilized Newton-Armijo Algorithm

Step 2. Go to Step 1, and compute $\nabla\Psi_{t_k}(x^{k,i}) = \nabla F_t(x^{k,i}) + C\nabla\Phi_t(x^{k,i})$. If $\|\nabla\Psi_{t_k}(x^{k,i})\|^2 > \tau(t_k)$, go to Step 3. Else go to Step 8.

Step 3. Compute $B_{t_k}(x^{k,i})$

$$B_{t_k}\left(x^{k,i}\right) = \theta\left(x^{k,i}\right)I + \nabla^2\Psi_{t_k}\left(x^{k,i}\right), \quad (21)$$

where $\theta(x) = \max\{0, \delta - e(x)\}$ with $e(x)$ denoting the minimum eigenvalue of $\nabla^2\Psi_{t_k}(x)$; then compute the Cholesky factor R such that $B_{t_k}(x^{k,i}) = RR^T$ and the reciprocal condition number $c(R)$ of R. If $c(R) \ge \kappa_1$ and $p_k \ge \kappa_3$, go to Step 4. Else, if $c(R) \ge \kappa_1$ and the largest eigenvalue $\sigma_{p_k,\max}(x^{k,i})$ of $B_{t_k}(x^{k,i})$ satisfies $\sigma_{t_k,\max}(x^{k,i}) \le \kappa_2$, go to Step 4; else go to Step 5.

Step 4. Compute the search direction

$$h_{k,i} = -B_{t_k}\left(x^{k,i}\right)^{-1}\nabla\Psi_{t_k}\left(x^{k,i}\right), \quad (22)$$

and go to Step 6.

Step 5. Compute the search direction

$$h_{k,i} = -\nabla\Psi_{t_k}\left(x^{k,i}\right). \quad (23)$$

Step 6. Compute the step length $\lambda_{k,i} = \beta^l$, where $l \ge 0$ is the smallest integer satisfying

$$\Psi_{t_k}\left(x^{k,i} + \beta^l h_{k,i}\right) - \Psi_{t_k}\left(x^{k,i}\right) \le \alpha\beta^l\left\langle\nabla\Psi_{t_k}\left(x^{k,i}\right), h_{k,i}\right\rangle. \quad (24)$$

Step 7. Set $x^{k,i+1} = x^{k,i} + \lambda_{k,i}h_{k,i}$, $i = i + 1$. Go to Step 1, and compute $\nabla\Psi_{t_k}(x^{k,i})$. If

$$\left\|\nabla\Psi_{t_k}\left(x^{k,i}\right)\right\|^2 \leq \tau\left(t_k\right), \tag{25}$$

go to Step 8; else go to Step 3.

Adaptive Parameter Decrease

Step 8. If $s = 1$, compute t^* such that

$$\epsilon_a\left(t_k\right) \leq \left\|\nabla\Psi_{t^*}\left(x^{k,i}\right)\right\|^2 \leq \epsilon_b\left(t_k\right), \tag{26}$$

go to Step 9, else set $t_{k+1} = 1/s(k+2)$, $k = k+1$, and $i = 0$, and go to Step 2.

Step 9. If $t^* \geq \hat{t}$, set $t_{k+1} = \min\{t^*, t_k/(t_k+1)\}$, $k = k+1$, and $i = 0$, and go to Step 2; else set $s = \max\{2, ((1/\hat{t})+2)/(k+1)\}$, $t_{k+1} = 1/s(k+2)$, $k = k+1$, and $i = 0$, and go to Step 2.

Lemma 7. *Suppose that Assumption 1 holds and that sequences $\{t_k\}$ and $\{x^{1,i}\}$, $\{x^{2,i}\}, \ldots, \{x^{k,i}\}, \ldots$, are generated by Algorithm 6. Then the following properties hold:* (1) *the sequences $\{t_k\}$ is decreasing;* (2) *If $\{x^{1,i}\}, \{x^{2,i}\}, \ldots, \{x^{k,i}\}, \ldots$ has an accumulation point, then $t_k \to 0$ and $\sum_{k=0}^{\infty} t_k = +\infty$.*

Lemma 8. *Suppose that Assumption 1 holds. Then, for every bounded set $S \subset R^n$ and parameters $\alpha, \beta \in (0,1)$, there exists a $K_s < \infty$ such that, for all $0 < t \leq 1$ and $x \in S$,*

$$\Psi_t\left(x + \lambda_t\left(x\right)h_t\left(x\right)\right) - \Psi_t\left(x\right) \leq -\alpha K_s\left\|\nabla\Psi_t\left(x\right)\right\|^2 t, \tag{27}$$

where $\lambda_t(x)$ is the stepsize of Algorithm 6 (see (24)*).*

Lemma 9. *Suppose that Assumption 1 holds and that $\{x^{1,i}\}, \{x^{2,i}\}, \ldots, \{x^{k,i}\}, \ldots$ is a bounded sequence generated by Algorithm 6. Then, for any k, the sequence $\{x^{k,i}\}$ is finite; that is, there exists a $i_k \in N$ such that* (25) *holds for $i = i_k$.*

The proofs of Lemmas 7–9 are similar to that in [23] and omitted here.

Theorem 10. *Suppose that Assumption 1 holds and that $\{x^{k,i_k}\}_{k=0}^{\infty}$ is a bounded sequence generated by Algorithm 6. Then, there exists an infinite subset $K \subset N$ and a $\hat{x} \in R^n$ such that $x^{k,i_k} \to^{K} \hat{x}$ and $0 \in \partial\psi(\hat{x})$.*

Proof. Suppose that $\{x^{k,i_k}\}_{k=0}^{\infty}$ is a bounded sequence generated by Algorithm 6. For the sake of a contradiction, suppose that there exists an $\varepsilon > 0$ such that

$$\liminf_{k\to\infty}\left\|\nabla\Psi_{t_k}\left(x^{k,i_k}\right)\right\| \geq \varepsilon. \tag{28}$$

Since $\{x^{k,i_k}\}_{k=0}^{\infty}$ is a bounded sequence, it has at least one accumulation point. Hence, by Lemma 7, $t_k \to 0$ as $k \to \infty$. Next, by Lemma 8, there exists an $M < \infty$ such that

$$\Psi_{t_k}\left(x^{k,i+1}\right) - \Psi_{t_k}\left(x^{k,i}\right) \leq -\alpha M\left\|\nabla\Psi_{t_k}\left(x^{k,i}\right)\right\|^2 t_k \tag{29}$$

for all $k \in N$. Hence, for all $k \in N$,

$$\begin{aligned}
&\Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}}\right) - \Psi_{t_k}\left(x^{k,i_k}\right)\\
&\quad = \left(\Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}}\right) - \Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}-1}\right)\right)\\
&\qquad + \left(\Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}-1}\right) - \Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}-2}\right)\right)\\
&\qquad + \cdots + \left(\Psi_{t_{k+1}}\left(x^{k+1,1}\right) - \Psi_{t_{k+1}}\left(x^{k+1,0}\right)\right)\\
&\qquad + \left(\Psi_{t_{k+1}}\left(x^{k+1,0}\right) - \Psi_{t_k}\left(x^{k,i_k}\right)\right)\\
&\quad \leq -\alpha M \sum_{j=0}^{i_{k+1}-1}\left\|\nabla\Psi_{t_{k+1}}\left(x^{k+1,j}\right)\right\|^2 t_{k+1},
\end{aligned} \tag{30}$$

where we have used the fact from Proposition 4 that $\Psi_{t_{k+1}}(x^{k+1,0}) - \Psi_{t_k}(x^{k,i_k}) \leq 0$ for all $k \in N$. In view of Algorithm 6, we know $x^{k+1,0} = x^{k,i_k}$. Then,

$$\begin{aligned}
&\Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}}\right)\\
&\quad = \left(\Psi_{t_{k+1}}\left(x^{k+1,i_{k+1}}\right) - \Psi_{t_k}\left(x^{k,i_k}\right)\right)\\
&\qquad + \left(\Psi_{t_k}\left(x^{k,i_k}\right) - \Psi_{t_{k-1}}\left(x^{k-1,i_{k-1}}\right)\right)\\
&\qquad + \cdots + \left(\Psi_{t_1}\left(x^{1,i_1}\right) - \Psi_{t_0}\left(x^{0,i_0}\right)\right) + \Psi_{t_0}\left(x^{0,i_0}\right)\\
&\quad \leq -\alpha M \sum_{j=0}^{i_{k+1}-1}\left\|\nabla\Psi_{t_{k+1}}\left(x^{k+1,j}\right)\right\|^2 t_{k+1}\\
&\qquad - \alpha M \sum_{j=0}^{i_k-1}\left\|\nabla\Psi_{t_k}\left(x^{k,j}\right)\right\|^2 t_k\\
&\qquad - \cdots - \alpha M \sum_{j=0}^{i_0-1}\left\|\nabla\Psi_{t_0}\left(x^{0,j}\right)\right\|^2 t_0 + \Psi_{t_0}\left(x^{0,i_0}\right).
\end{aligned} \tag{31}$$

By Lemma 7, $\sum_{k=0}^{\infty} t_k = +\infty$. It follows from (28) and (31) that $\Psi_{t_k}(x^{k,i_k}) \to -\infty$, as $k \to \infty$. But for every accumulation point x^* of $\{x^{k,i_k}\}$, that is, there exists an infinite subset $K^* \subset N$ such that $x^{k,i_k} \to^{K^*} x^*$, we have by continuity $\Psi_{t_k}(x^{k,i_k}) \to^{K^*} \Psi(x^*)$, which is a contradiction. Hence,

$$\liminf_{k\to\infty}\left\|\nabla\Psi_{t_k}\left(x^{k,i_k}\right)\right\| = 0. \tag{32}$$

Now by (16) and (18) and Proposition 3.2 in [25], we have that, for all $x \in R^n$,

$$\begin{aligned}
\lim_{k\to\infty}\nabla\Psi_t\left(x^{k,i_k}\right) &= \sum_{j\in s(\hat{x})}\hat{\lambda}_j\left(\hat{x}\right)\nabla f_j\left(\hat{x}\right)\\
&\quad + C\sum_{i\in q(\hat{x})}\hat{\xi}_i\left(\hat{x}\right)\nabla\phi_i\left(\hat{x}\right) = 0.
\end{aligned} \tag{33}$$

According to Proposition 3.2 in [25], we have $\widehat{\lambda}_j(\widehat{x}), \widehat{\xi}_i(\widehat{x}) \geq 0$, $j \in p(\widehat{x})$, and $i \in q(\widehat{x})$ such that $\sum_{j\in p(\widehat{x})} \widehat{\lambda}_j(\widehat{x}) = 1$, $\sum_{i\in q(\widehat{x})} \widehat{\xi}_i(\widehat{x}) = 1$. Hence by (32) and (33), there has to exist an infinite subset $K \subset N$ and $\widehat{x} \in R^n$ such that $x^{k,i_k} \rightarrow^K \widehat{x}$ and $0 \in \partial\psi(\widehat{x})$. This completes the proof. □

3. Numerical Experiment

In this section, we consider the rotated cone fitting problem in [27]. In this example, the error δ_j associated with u_j is zero, and $u_j = (u_j^1, u_j^2)$. Let $(u_j^1, u_j^2$, and $w_j) \in R^3$, $i = 1, \ldots, m$, be the measured data. Define the orthogonal matrix

$$T(x_4, x_5) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(x_4) & \sin(x_4) \\ 0 & -\sin(x_4) & \cos(x_4) \end{bmatrix} \times \begin{bmatrix} \cos(x_5) & 0 & -\sin(x_5) \\ 0 & 1 & 0 \\ \sin(x_5) & 0 & \cos(x_5) \end{bmatrix}. \tag{34}$$

Then $(u_j^1, u_j^2$, and $w_j)$ is transformed to $(\overline{u}_j^1, \overline{u}_j^2$, and $\overline{w}_j) = ((u_j^1 + x_1), (u_j^2 + x_2), (w_j + x_3))T(x_4, x_5)$. The fitting cone V is defined as $\overline{w} = x_6\sqrt{(\overline{u}_j^1)^2 + (\overline{u}_j^2)^2}$, $x_6 \geq 0$. Denote the unknown variables to be $x = (x_1, x_2, x_3, x_4, x_5$, and $x_6)$; then the fitting problem is equivalent to

$$\begin{aligned} \min \quad & f(x) = \max_{1\leq j\leq 2m} \{f_j(x)\}, \\ \text{s.t.} \quad & x \in \Lambda = \{x \mid -\pi \leq x_4 \leq \pi, -\pi \leq x_5 \leq \pi, 0 \leq x_6\}, \end{aligned} \tag{35}$$

where

$$f_j(x) = \begin{cases} \overline{w}_j(x) - x_6\sqrt{\left(\overline{u}_j^1(x)\right)^2 + \left(\overline{u}_j^2(x)\right)^2} & j = 1, \ldots, m \\ -f_{j-m}(x) & j = m+1, \ldots, 2m. \end{cases} \tag{36}$$

Since $f_j(x)$ in (35) is nonsmooth in $\mathfrak{R}_j = \{x \mid (\overline{u}_j^1(x))^2 + ((\overline{u}_j^2)(x))^2 = 0\}$, we try to smooth it by the following function:

$$f_{j,t}(x) = \begin{cases} \overline{w}_j(x) - x_6\sqrt{\left(\overline{u}_j^1(x)\right)^2 + \left(\overline{u}_j^2(x)\right)^2 + t} + x_6\sqrt{t}, \\ \qquad\qquad j = 1, \ldots, m, \\ x_6\sqrt{\left(\overline{u}_j^1(x)\right)^2 + \left(\overline{u}_j^2(x)\right)^2 + t} - \overline{w}_j(x), \\ \qquad\qquad j = m+1, \ldots, 2m. \end{cases} \tag{37}$$

Let $\phi(x) = \max\{\phi_1(x), \ldots, \phi_6(x)\}$, where $\phi_1(x) = x_4 - \pi$, $\phi_2(x) = -x_4 - \pi$, $\phi_3(x) = x_5 - \pi$, $\phi_4(x) = -x_5 - \pi$, $\phi_5(x) = -x_6$, $\phi_6(x) = 0$.

Use penalty function with penalty parameter $C > 0$ to transform (35) into the following unconstrained programming problem:

$$\min \left\{\psi(x) = \left\{\max_{1\leq j\leq 2m} f_{j,t}(x) + C\phi(x)\right\}\right\}. \tag{38}$$

According to the definition of $f_{j,t}(x)$, it is easy to obtain the following proposition.

Proposition 11. *For any $t > 0$, $f_{j,t}(x)$ defined as (37) is twice continuously differentiable, and for any given $x \in \Lambda$, $f_{j,t}(x)$ is monotonically increasing with respect to $t > 0$, and $f_{j,t}(x) \rightarrow f_j(x)$ as $t \rightarrow 0$.*

Remark 12. Under Proposition 11, $F_t(x)$ is monotonically increasing with respect to $t > 0$.

Remark 13. Suppose that $\{x^{k,i_k}\}_{k=0}^{\infty}$ is a bounded sequence generated by Algorithm 6. Then, there exists an infinite subset $K \subset N$ and a $\widehat{x} \in R^6$ such that $x^{k,i_k} \rightarrow^K \widehat{x}$ and

$$0 \in \underset{j\in s(\widehat{x})}{\text{conv}} \left\{\partial f_j(\widehat{x})\right\} + C \underset{i\in q(\widehat{x})}{\text{conv}} \left\{\partial\phi_i(\widehat{x})\right\}. \tag{39}$$

Moreover, if $\widehat{x} \notin \mathfrak{R} = \bigcup_{1\leq j\leq 2m} \mathfrak{R}_j$, then $0 \in \partial\psi(\widehat{x})$.

We have implemented the SN algorithm using the MATLAB for problem (38). In order to show the efficiency of the algorithm, we also have implemented TSN algorithms using similar procedures and an SQP algorithm that is implemented by calling MATLAB function fminimax directly. Algorithm TSN was proposed by Xiao et al. in [27].

The test results were obtained by running MATLAB R2011a on a desktop with Windows XP Professional operation system, Intel(R) Core(TM) i3-370 2.40 GHz processor and 2.92 GB of memory. The default parameters are chosen as follows:

$$\begin{gathered} \alpha = 0.8, \qquad \beta = 0.77, \qquad \widehat{t} = 10^5 \ln(2m), \\ \kappa_1 = 10^{-7}, \qquad \kappa_2 = 10^{30}, \qquad \kappa_3 = 1000\widehat{t}, \\ \tau(t) = 10^{-3}, \qquad t_0 = 1, \qquad (\epsilon_a, \epsilon_b) = (0.01, 0.2), \\ \delta = 0.1, \qquad C = 100. \end{gathered} \tag{40}$$

The results are listed in Table 1, where x^* denotes the final approximate solution point and $f(x^*)$ is the value of the objective function at x^*. Time is the CPU time in seconds.

We test our algorithm for the artificial rotated cone data points which are generated as that in [28]. At first, produce points $\{(\widetilde{z}_1^j, \widetilde{z}_2^j$, and $\widetilde{z}_3^j)\}_{j=1}^m$ on an unrotated cone by defining the $\widetilde{z}_1^j = r_j\tan(\pi/6)\cos(\gamma_j)$, $\widetilde{z}_2^j = r_j\tan(\pi/6)\sin(r_j)$, $\widetilde{z}_3^j = r_j$, where r_j and γ_j are equally distributed pseudorandom numbers in $[1; 10]$ and $[0; 2\pi]$, respectively. Then, perturb $\widetilde{z}_3^j$ by adding error item which follows the Gaussian distribution $N(0; 0.3)$, and make rotations and translation to obtain the final data $((u_j^1 + 2.1), (u_j^2 - 1.4), (w_j + 1.3)) = (\widetilde{z}_1^j, \widetilde{z}_2^j, \widetilde{z}_3^j)T^{-1}(\pi/20, \pi/25)$.

TABLE 1: Test result for the example, $x^0 = (0, 0, -2, 2, -2, 1)$, $t = 10^{-5}$.

m	Method	x^*	$f(x^*)$	Time
	SN	$(-0.156746, -0.124595, \ldots, 1.732075)$	0.744533	1.891536
20000	TSN	$(-0.156611, -0.123928, \ldots, 1.736362)$	0.743643	3.878818
	SQP	$(-0.155824, -0.126803, \ldots, 1.760751)$	0.927184	31.302830
	SN	$(-0.156984, -0.125677, \ldots, 1.732449)$	0.851957	4.166439
50000	TSN	$(-0.155253, -0.123374, \ldots, 1.733202)$	0.851803	10.761201
	SQP	$(-0.155824, -0.126194, \ldots, 1.760880)$	0.971305	33.716271
	SN	$(-0.155158, -0.124954, \ldots, 1.736343)$	0.856780	16.437595
100000	TSN	$(-0.155159, -0.124956, \ldots, 1.736346)$	0.856767	37.235130
	SQP	$(-0.157418, -0.125016, \ldots, 1.761221)$	1.025103	101.817872

Acknowledgments

This work was supported by the National Natural Science Foundation of China (11171051, 91230103, and 61153001) and the Fundamental Research Funds for the Central Universities of China (DC110305).

References

[1] P. T. Boggs, R. H. Byrd, and R. B. Schnabel, "A stable and efficient algorithm for nonlinear orthogonal distance regression," *SIAM Journal on Scientific Computing*, vol. 8, pp. 1052–1078, 1987.

[2] R. Strebel, D. Sourlier, and W. Gander, "A comparison of orthogonal least squares fitting in coordinate metrology," in *Recent Advances in Total Least Squares and Errors-in-Variables Techniques*, S. Van Huffel, Ed., pp. 249–258, SIAM, Philadelphia, Pa, USA, 1997.

[3] J. W. Zwolak, P. T. Boggs, and L. T. Watson, "Algorithm 869: ODRPACK95: a weighted orthogonal distance regression code with bound constraints," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, Article ID 1268782, 2007.

[4] I. Al-Subaihi and G. A. Watson, "Fitting parametric curves and surfaces by l_∞ distance regression," *BIT Numerical Mathematics*, vol. 45, no. 3, pp. 443–461, 2005.

[5] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer, New York, NY, USA, 1985.

[6] W. Murray and L. M. Overton, "A projected Lagrangian algorithm for nonlinear minimax optimization," *SIAM Journal on Scientific Computing*, vol. 1, pp. 345–370, 1980.

[7] R. S. Womersley and R. Fletcher, "An algorithm for composite nonsmooth optimization problems," *Journal of Optimization Theory and Applications*, vol. 48, no. 3, pp. 493–523, 1986.

[8] J. L. Zhou and A. L. Tits, "An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 461–487, 1996.

[9] A. Frangioni, "Generalized bundle methods," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 117–156, 2003.

[10] M. Gaudioso and M. F. Monaco, "A bundle type approach to the unconstrained minimization of convex nonsmooth functions," *Mathematical Programming*, vol. 23, no. 2, pp. 216–226, 1982.

[11] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, vol. 1133 of *Lecture Notes in Mathematics*, Springer, Berlin, Germany, 1985.

[12] J. Zowe, "Nondifferentiable optimization: a motivation and a short introduction into the sub-gradient and the bundle concept," in *Computational Mathematical Programming*, K. Schittkowski, Ed., vol. 15 of *NATO SAI Series*, pp. 323–356, Springer, New York, NY, USA, 1985.

[13] J. W. Bandler and C. Charalambous, "Practical least pth optimization of networks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 20, no. 12, pp. 834–840, 1972.

[14] C. Charalambous, "Acceleration of the least pth algorithm for minimax optimization with engineering applications," *Mathematical Programming*, vol. 17, no. 1, pp. 270–297, 1979.

[15] C. Gfgola and S. Gomez, "A regularization method for solving the finite convex min-max problem," *The SIAM Journal on Numerical Analysis*, vol. 27, pp. 1621–1634, 1990.

[16] D. Q. Mayne and E. Polak, "Nondifferential optimization via adaptive smoothing," *Journal of Optimization Theory and Applications*, vol. 43, no. 4, pp. 601–613, 1984.

[17] E. Polak, J. O. Royset, and R. S. Womersley, "Algorithms with adaptive smoothing for finite minimax problems," *Journal of Optimization Theory and Applications*, vol. 119, no. 3, pp. 459–484, 2003.

[18] R. A. Polyak, "Smooth optimization methods for minimax problems," *SIAM Journal on Control and Optimization*, vol. 26, no. 6, pp. 1274–1286, 1988.

[19] S. Xu, "Smoothing method for minimax problems," *Computational Optimization and Applications*, vol. 20, no. 3, pp. 267–279, 2001.

[20] I. Zang, "A smoothing technique for minCmax optimization," *Mathematics Programs*, vol. 19, pp. 61–77, 1980.

[21] Y. Xiao and B. Yu, "A truncated aggregate smoothing Newton method for minimax problems," *Applied Mathematics and Computation*, vol. 216, no. 6, pp. 1868–1879, 2010.

[22] G. H. Zhao, Z. R. Wang, and H. N. Mou, "Uniform approximation of min/max functions by smooth splines," *Journal of Computational and Applied Mathematics*, vol. 236, pp. 699–703, 2011.

[23] L. Dong and B. Yu, "A spline smoothing Newton method for finite minimax problems," preprint.

[24] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Springer, New York, NY, USA, 1997.

[25] L. Dong, B. Yu, and G. H. Zhao, "A smoothing spline homotopy method for nonconvex nonlinear programming," preprint.

[26] B. Yu, G. C. Feng, and S. L. Zhang, "The aggregate constraint homotopy method for nonconvex nonlinear programming," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 45, no. 7, pp. 839–847, 2001.

[27] Y. Xiao, B. Yu, and D. L. Wang, "Truncated smoothing newton method for l_∞ fitting rotated cones," *Journal of Mathematical Research and Exposition*, vol. 30, pp. 159–166, 2010.

[28] H. Späth, "Least squares fitting with rotated paraboloids," *Mathematical Communications*, vol. 6, no. 2, pp. 173–179, 2001.