Fitting Two Concentric Circles and Spheres to Data by $l_1$ Orthogonal Distance Regression

I. A. Al-Subaihi

Department of Science, King Abdulaziz Military Academy
P.O. Box 73052, Riyadh 11538, Saudi Arabia
alsubaihi@hotmail.com

Abstract

The problem of fitting two concentric circles and spheres to data arise in computational metrology. The most commonly used criterion for this is the Least Square norm. There is also interest in other criteria, and here we focus on the use of the $l_1$ norm, which is traditionally regarded as important when the data contain wild points. A common approach to this problem involves an iteration process which forces orthogonality to hold at every iteration and steps of Gauss-Newton type. However, a simple and robust algorithm assigned for using the $l_1$ Orthogonal Distance Regression will be proposed in this paper. The efficient algorithm is demonstrated by some numerical example.

Mathematics Subject Classification: 65D10

Keywords: Two concentric circles (spheres), orthogonal distance regression, $l_1$ norm, Gauss-Newton method

1 Introduction

To find a measured points in plane (or space) two concentric circles (or spheres) of minimum difference in radii that contain all points between them is a problem in computational metrology [1, 10, 12], global differential geometry, model theory, and in many domains within the field of computer vision, see for examples [7, 18].

In order to make the problem more evident, it is recommended to assume that all points $x_i \in R^s$ have errors, where $i = 1, \ldots, m$ and $s = 2$ or 3. Moreover, to find two concentric circles (spheres), it is suggested that such suitable criteria goes to its minimum level, where simultaneously each point has to be associated with one of the circles (spheres), see for examples [1, 10].
The parametric representations are dependent upon the location parameter \( \alpha \), which are independent of any coordinate system [13]. The small circle (sphere) of two concentric circles (spheres) can be given in parametric representation as

\[
x(a_1, \alpha) = c + r_1 \beta
\]

where \( a_1 = (c, r_1)^T \), \( r_1 \) the radius, \( c \in \mathbb{R}^s \), \( s = 2 \) or \( 3 \), is the common centre, and where

\[
\beta_i = \begin{bmatrix}
    \cos(\alpha_i)
    \\
    \sin(\alpha_i)
\end{bmatrix},
\]

for circles, and

\[
\beta_i = \begin{bmatrix}
    \cos(\alpha_i) \sin(\alpha_{i2})
    \\
    \sin(\alpha_i) \sin(\alpha_{i2})
    \\
    \cos(\alpha_{i2})
\end{bmatrix},
\]

for spheres representations. Replacing \( a_1 \) by \( a_2 = (c, r_2)^T \) in (1) gives the large concentric circle (sphere) with \( r_2 \) radius and \( r_2 > r_1 \).

Fitting two concentric spheres to data is interpreted as a linear Least Squares problem in [10], which seems to have been the first suggested. Let the disjoint sets \( V_1 \) and \( V_2 \) be sets of indices of given points associated with the small and large concentric circles (spheres) respectively. Moreover, \( V_1 \cup V_2 = \{1, \ldots, m\} \), \( |V_1| \geq 4 \), \( |V_2| \geq 4 \). Then the problem in [10], in brief, is to find two sets \( V_1 \) and \( V_2 \) of indices of points such that the sum of squared minimal distance

\[
S(V_1, V_2, a) = \sum_{i \in V_1} \min_{\alpha} v_1^2(a_1, \alpha) + \sum_{i \in V_2} \min_{\alpha} v_2^2(a_2, \alpha)
\]

is minimized, where \( a = (c, r_1, r_2)^T \in \mathbb{R}^5 \), and where

\[
v_1^2(a_1, \alpha_i) = \sum(x_i - x(a_1, \alpha_i))^2, \quad i \in V_1,
\]

and \( v_2^2(a_2, \alpha_i), \quad i \in V_2 \) represent the large sphere.

Two concentric spheres fitting using the Orthogonal Distance Regression or ODR has been studied by [1]. Let \( u_{1i} = x_i - x(a_1, \alpha_i), i \in V_1 \) and \( u_{2i} = x_i - x(a_2, \alpha_i), i \in V_2 \). Then the ODR problem in [1] was to find the two sets \( V_1 \) and \( V_2 \) such minimizing

\[
F(V_1, V_2, a, \alpha) = \sum_{i \in V_1} ||u_{1i}(a_1, \alpha_i)||^2 + \sum_{i \in V_2} ||u_{2i}(a_2, \alpha_i)||^2,
\]

with respect to \( a = (c, r_1, r_2)^T \in \mathbb{R}^5 \) and \( \alpha_i, \quad i = 1, \ldots, m \). Where in all cases in this paper, \( ||.|| \) denotes the \( l_2 \) vector or matrix norm.

While Least Squares problems are dominated. There is interested in other criteria. For example the \( l_\infty \) norm has a role to play in the context of accept/reject tests, see for examples [2, 4, 9]. The \( l_1 \) norm is traditionally regarded as important in the presence of outliers, see for examples [2, 3, 4, 8, 14,
Fitting two concentric circles and spheres

1023

The purpose in this paper, is to fit two concentric circles (spheres) to data by \( l_1 \) orthogonal Orthogonal Distance Regression (\( l_1 \) ODR) problem. The \( l_1 \) ODR problem will be defined in next section. Further, The Jacobian matrix for the Gauss-Newton iteration is defined in section 3. Furthermore, an algorithm and the starting points are considered in section 4. Numerical results are considered in sections 5.

2 The \( l_1 \) ODR problem

The problem given in this paper is to find two sets \( V_1 \) and \( V_2 \) by minimizing

\[
G(V_1, V_2, \mathbf{a}, \mathbf{\alpha}) = \sum_{i \in V_1} \| \mathbf{u}_{1i}(\mathbf{a}_1, \mathbf{\alpha}_i) \| + \sum_{i \in V_2} \| \mathbf{u}_{2i}(\mathbf{a}_2, \mathbf{\alpha}_i) \|,
\]

with respect to \( \mathbf{a} \in \mathbb{R}^n \), \( n = 4 \) for fitting two concentric circles and \( n = 5 \) for two concentric spheres, and \( \mathbf{\alpha} \in \mathbb{R}^m \).

Let \( \mathbf{\alpha}_i(\mathbf{a}_1) \), \( i \in V_1 \) be such that for any \( \mathbf{a}_1 \), \( \| \mathbf{x}_i - \mathbf{x}(\mathbf{a}_1, \mathbf{\alpha}_i) \| \) is minimized with respect to \( \mathbf{\alpha}_i \), \( i \in V_1 \), and similarly for \( \mathbf{\alpha}_i(\mathbf{a}_2) \), \( i \in V_2 \), see [1, 2, 4, 5, 8, 14]. Then the equivalent problem to (4) is the minimization with respect to \( \mathbf{a} = (\mathbf{c}, r_1, r_2)^T \) alone of

\[
G(V_1, V_2, \mathbf{a}) = \sum_{i \in V_1} \| \mathbf{u}_{1i}(\mathbf{a}_1, \mathbf{\alpha}_i(\mathbf{a}_1)) \| + \sum_{i \in V_2} \| \mathbf{u}_{2i}(\mathbf{a}_2, \mathbf{\alpha}_i(\mathbf{a}_2)) \|.
\]

If we define, for simplicity, the vector \( \mathbf{\delta} \in \mathbb{R}^m \) which has \( i \)th component

\[
\delta_i(\mathbf{a}) = \| \mathbf{x}_i - \mathbf{c} - r_1 \mathbf{\beta}_i - r_2 \mathbf{\beta}_i \|,
\]

and \( r_1 = 0 \) if \( i \in V_2 \), \( r_2 = 0 \) if \( i \in V_1 \). Then the \( l_1 \) Orthogonal Distance Regression (\( l_1 \) ODR) problem is to find two sets \( V_1 \) and \( V_2 \) such as minimizing the objective function

\[
G(V_1, V_2, \mathbf{a}) = \| \mathbf{\delta}(\mathbf{a}) \|_1 = \sum_{i=1}^{m} \delta_i.
\]

This problem is not convex in \( \mathbf{a} \). As is normal in non-convex problems, it is necessary to satisfy the first order conditions for a minimum, or to find a stationary point. Such a point may be at best a local minimizer [1, 2, 4, 6, 8, 14, 15, 17].

**Definition 1** [17, 15] \( \mathbf{a} \) is a stationary point of (7) if there exists the subgradients \( \mathbf{w} \in \mathbb{R}^n \), where \( |w_i| \leq 1, \delta_i = 0; w_i = \text{sign}(\delta_i), \delta_i \neq 0 \) such that

\[
\nabla_{\mathbf{a}} \mathbf{\delta}^T \mathbf{w} = 0,
\]
where $\nabla_a \delta$ is the Jacobian matrix at the vector $a$ given by
\[
\nabla_a \delta = \frac{\partial \delta_i}{\partial a_j}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.
\]

The most popular method for solving problems as (7) is the Gauss-Newton method or one of its variants, see for examples [2, 3, 4, 8, 14, 15, 16, 17].

3 A Gauss-Newton step $d$

The Gauss-Newton step $d$ for minimizing (7) is given by finding
\[
\min_d \| \delta + \nabla_a \delta d \|_1,
\]
where $\nabla_a \delta \in \mathbb{R}^{m \times s}$ is the Jacobian matrix which has $i$th row, for any $\delta_i \neq 0$
\[
\nabla_a \delta_i = \begin{cases} 
-\frac{u_{1i}^T}{\|u_1\|} \begin{bmatrix} I & \beta_i & 0 \end{bmatrix} & \text{for } i \in V_1,
-\frac{u_{2i}^T}{\|u_2\|} \begin{bmatrix} I & 0 & \beta_i \end{bmatrix} & \text{for } i \in V_2,
\end{cases}
\]
by definition of $\alpha_i(a)$, and $I$ signifies the identity matrix. Then the correct vector of partial derivatives of $\delta_i$ is easily calculate.

The performance of the Gauss-Newton methods for (7), depends primarily on the number of zero components of $\delta$, at a limit point: if this number is $n$, then a second order rate of convergence is expected. As a result of the connection with the Newton method which is applied to the $n$ nonlinear equations
\[
\delta_i(a) = 0, \quad i \in \bar{I}, \quad |\bar{I}| = n,
\]
satisfied at the stationary point, and the same directions in the both linear systems, see for examples [2, 5, 6, 8, 14, 16, 17].

4 An algorithm

The contents of the last two sections give the ingredients of algorithms which can be used to solve the $l_1$ Orthogonal Distance Regression ($l_1$ ODR) problem. For example, we could consider an algorithm as follows

**STEP 0.** Input: The data $x_i$, the iteration number $k = 0$ and a tolerance ($Tol$).

**STEP 1.** Determine: the initial values of $a^{(0)}$, $V_1^{(0)}$ and $V_2^{(0)}$.

It is necessary to provide primary values for any iterative algorithm. The initial centre $c$ can be determined by the data mean. The primary values
assigned for radii can be determined by calculating the minimal and maximal distance of the data points to the starting centre respectively. Further, a good guess can be used as initial values of sets $V_1$ and $V_2$. Actually, a random choosing is used in the examples, as we will see in section 5.

**STEP 2.** Determine: $\alpha_i(a^{(k)})$, $i = 1, \ldots, m$.

The Least Squares solution is used to determine the values of $\alpha_i(a^{(k)})$, $i = 1, \ldots, m$, see for example [10, 11], the algorithm [1]. For $k$th iteration

$$\alpha_{i1} = \arctan \left| \frac{y_i - c_2}{x_i - c_1} \right|,$$

- If $x_i - c_1 < 0$, then $\alpha_{i1} = \alpha_{i1} + \pi$.
- If $x_i - c_1 = 0$, then
  - $\alpha_{i1} = \frac{\pi}{2}$, if $y_i - c_2 \geq 0$.
  - $\alpha_{i1} = \frac{3}{2}\pi$, if $y_i - c_2 < 0$.

$$\alpha_{i2} = \arctan \left| \frac{(x_i - c_1) \cos(\alpha_{i1}) + (y_i - c_2) \sin(\alpha_{i1})}{z_i - c_3} \right|$$

- If $z_i - c_3 < 0$, then $\alpha_{i2} = \alpha_{i2} + \pi$.
- If $z_i - c_3 = 0$, then
  - $\alpha_{i2} = \frac{\pi}{2}$, if $(x_i - c_1) \cos(\alpha_{i1}) + (y_i - c_2) \sin(\alpha_{i1}) > 0$.
  - $\alpha_{i2} = \frac{3}{2}\pi$, if $(x_i - c_1) \cos(\alpha_{i1}) + (y_i - c_2) \sin(\alpha_{i1}) < 0$.

**STEP 3.** Find:

- $\delta(a^{(k)})$ and (6).
- The objective function $G^{(k)}$ and (7).
- The Jacobian matrix $\nabla_{a^{(k)}} \delta(a^{(k)})$ and (8).

**STEP 4.** Solve: $\|\delta(a^{(k)}) + \nabla_{a^{(k)}} \delta(a^{(k)}) d^{(k)}\|_1$ for $d^{(k)}$.

- If $\|d^{(k)}\|_\infty < \text{Tol}$, then Go to Step 8.

- Otherwise use line search.
  Calculate a new estimate $a^{(k+1)} = a^{(k)} + \gamma d^{(k)}$.

**STEP 5.** Set $k = k + 1.$
STEP 6. Determine: $V_1^{(k)}$ and $V_2^{(k)}$.

The disjoint sets $V_1^{(k)}$ and $V_2^{(k)}$ can be determined by calculating, for each $i$, $\|u_1^{(k)}(i)\|_1$ and $\|u_2^{(k)}(i)\|_1$. If $\|u_1^{(k)}(i)\|_1 < \|u_2^{(k)}(i)\|_1$, then $i \in V_1^{(k)}$ otherwise in $V_2^{(k)}$.

STEP 7. Go to STEP 2.

STEP 8. The process of fitting the two concentric circles (spheres) using the $l_1$ ODR has been completed.

5 Numerical Results

In order to verify the performance of the proposed algorithm, it is applied to some examples. Therefore, the plan adopted in this regard, is to compute a Gauss-Newton step $d$ as indicated in the last two sections, to be used as an approach to reduce the value of the objective function $G$, taking full steps ($\gamma = 1$), if possible. The performance shown in the examples is typical, with a second order rate of convergence usual.

The first set, of size $|V_1|$, of indices of data points are generated by taking a particular circle (sphere). The second set, of size $|V_2|$, also generated in the same idea and same centre, but $r_2 > r_1$. In other word, the data has been generated by the two concentric circles (spheres) of coefficient $\hat{a}$. To make the comparison of the example easy, we let $|V_1| = |V_2| = m/2$. Then random perturbations are introduced for these data. The initial sets $V_1^{(0)}$ and $V_2^{(0)}$ with different sizes have been determined by taking random permutation of the integers from 1 to $m$, and the Matlab command ”randperm” is used for this procedure. The symbol $k$ is used in the following tables to refer the iteration number.

Example 1 Consider data points $x_i \in R^2$, $i = 1, \ldots, 20$ generated by the two concentric circles of coefficient $\hat{a} = (2, 3, 1.5, 2)^T$. The points $x_8, x_{13}$ are made to be wild points. The Gauss-Newton method is applied with the line searching for fitting these points with two concentric circles. The initial value of

$$
a^{(0)} = (2.1127, 2.9726, 6.1312, 3.3027)^T.
$$

Assume $V_2^{(0)} = \{7, 9, 10, 11, 12, 19\}$ and $V_1^{(0)}$ is the complement set of indices of the data points, which means that the two wild points in $V_1^{(0)}$. The method gives $V_1 = \{1, 2, \ldots, 10\}$ and $V_2 = \{11, 12, \ldots, 20\}$ in the first iterations. Further, the set of indices of zero components of $\delta_i$ is $\bar{I} = \{7, 12, 14, 17\}$. Results
are shown in Table 1. The assumption that $V_2^{(0)} = \{7, 9, 11, 12, 13, 19\}$, gives the final solution

$$a^{(4)} = (2.0254, 3.0919, 1.5830, 2.0237)^T,$$

with objective function $G = 6.0241$ and $\|d\|_\infty = 1.4852 \times 10^{-7}$. Further, the number of zero components of $\delta$ is 4, which has the indices $\bar{I} = \{4, 7, 16, 18\}$.

**Example 2** We fit two concentric circles to $m = 100$ points, where $\hat{a} = (2, 3, 1.5, 2)^T$, and 5 wild point are made. The starting sets $V_1^{(0)} \in R^{70}$ and $V_2^{(0)} \in R^{30}$ are also randomly permuted. The initial approximation to $a^{(0)}$ gives

$$a^{(0)} = (5.1554, 4.0464, 2.8048, 5.6668)^T.$$

Typical performance is shown in Table 6. The sets $V_1$ and $V_2$ have been, with $V_1 = \{1, 2, \ldots, 50\}$, $V_2 = \{51, 52, \ldots, 100\}$ in the first iteration, and the number of zero components of $\delta$ is 4, and $\bar{I} = \{14, 70, 85, 97\}$.

**Example 3** Next we consider the example from [10], to fit two concentric spheres. A vector of ones has been subtracted from $x_4$ to make it wild point, as shown in Table 3. The index of the wild point has been assumed in the set $V_2^{(0)}$. Starting value of $a^{(0)}$ gives

$$a^{(0)} = (1.6250, 2.1250, 1.2500, 2.2845, 6.9440)^T.$$

Applying the Gauss-Newton method with line search for fitting the 16 points by two concentric spheres. Table 4 shows the characterization of $V_1$ and $V_2$ during the iterations, where an integer vector $f$ with $f_i = 1$ if $i \in V_1$ and $f_i = 2$ if $i \in V_2$. The method gives $V_1 = \{1, 2, \ldots, 8\}$ and $V_2 = \{11, 12, \ldots, 16\}$ in 2 iterations. Results are shown in Table 5. The number of zero components of $\delta$ should be no less than $n = 5$, for second rate of convergence. In addition, the set of zero indices $i, \delta_i = 0$ is $\bar{I} = \{1, 5, 8, 9, 10, 12\}$.

**Example 4** As a final example we fit two concentric spheres to 100 points, where $|V_1| = |V_2| = 50$ and $\hat{a} = (-2, 3, 1, 6, 7)^T$. The initial sets were $V_1^{(0)} \in R^{70}$ and $V_2^{(0)} \in R^{30}$ is randomly chosen. The initial approximation to $a^{(0)}$ gives

$$a^{(0)} = (-2.3013, 2.8576, -0.4054, 5.5048, 9.8653)^T.$$

Results for fitting two concentric spheres to 100 points are shown in Table 6. The sets $V_1$ and $V_2$ have been, with $V_1 = \{1, 2, \ldots, 50\}$ and $V_2 = \{51, 52, \ldots, 100\}$ in the second iteration and $\bar{I} = \{1, 46, 50, 71, 83\}$. 
Table 1: Example 1.

<table>
<thead>
<tr>
<th>k</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$G$</th>
<th>$|d|_\infty$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1506</td>
<td>3.1076</td>
<td>1.5381</td>
<td>1.9672</td>
<td>6.0638</td>
<td>1.0052</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2.0371</td>
<td>3.1474</td>
<td>1.5371</td>
<td>2.0408</td>
<td>6.0241</td>
<td>1.1344</td>
<td>$\times 10^{-1}$</td>
</tr>
<tr>
<td>3</td>
<td>2.0372</td>
<td>3.1474</td>
<td>1.5405</td>
<td>2.0431</td>
<td>6.0126</td>
<td>3.4338</td>
<td>$\times 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>2.0372</td>
<td>3.1474</td>
<td>1.5405</td>
<td>2.0431</td>
<td>6.0126</td>
<td>1.3275</td>
<td>$\times 10^{-10}$</td>
</tr>
</tbody>
</table>

Table 2: Example 2.

<table>
<thead>
<tr>
<th>k</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$G$</th>
<th>$|d|_\infty$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.2289</td>
<td>3.2560</td>
<td>4.2033</td>
<td>4.7186</td>
<td>59.894</td>
<td>1.3985</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4.2161</td>
<td>3.2561</td>
<td>4.1319</td>
<td>4.9368</td>
<td>49.982</td>
<td>2.1817</td>
<td>$\times 10^{-1}$</td>
</tr>
<tr>
<td>3</td>
<td>4.1724</td>
<td>3.2535</td>
<td>4.1322</td>
<td>4.9149</td>
<td>49.901</td>
<td>4.3697</td>
<td>$\times 10^{-2}$</td>
</tr>
<tr>
<td>4</td>
<td>4.1851</td>
<td>3.2571</td>
<td>4.1384</td>
<td>4.9073</td>
<td>49.887</td>
<td>1.2672</td>
<td>$\times 10^{-2}$</td>
</tr>
<tr>
<td>5</td>
<td>4.1851</td>
<td>3.2571</td>
<td>4.1351</td>
<td>4.9073</td>
<td>49.887</td>
<td>3.3646</td>
<td>$\times 10^{-3}$</td>
</tr>
<tr>
<td>6</td>
<td>4.1851</td>
<td>3.2571</td>
<td>4.1351</td>
<td>4.9073</td>
<td>49.887</td>
<td>1.8745</td>
<td>$\times 10^{-12}$</td>
</tr>
</tbody>
</table>
Figure 2: Fitting two concentric circles to 100 data points.

Table 3: Example 3 data points

<table>
<thead>
<tr>
<th>x</th>
<th>1</th>
<th>-1</th>
<th>0</th>
<th>-2</th>
<th>2</th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>1</th>
<th>-1</th>
<th>0</th>
<th>-1</th>
<th>3</th>
<th>2</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>6</td>
<td>4</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>7</td>
<td>-2</td>
<td>-2</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4: $f$ during the iterations

<table>
<thead>
<tr>
<th>$k$</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 5: Example 3.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$G$</th>
<th>$|d|_\infty$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.2310</td>
<td>2.7726</td>
<td>2.5196</td>
<td>4.3705</td>
<td>4.6965</td>
<td>7.2287</td>
<td>2.2474</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.4461</td>
<td>2.7016</td>
<td>3.0386</td>
<td>4.1610</td>
<td>5.1664</td>
<td>3.4685</td>
<td>$7.8486 \times 10^{-1}$</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1.4992</td>
<td>2.9978</td>
<td>3.0009</td>
<td>4.1425</td>
<td>5.1149</td>
<td>1.5850</td>
<td>$2.9618 \times 10^{-1}$</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1.5000</td>
<td>3.0000</td>
<td>3.0000</td>
<td>4.1533</td>
<td>5.1235</td>
<td>1.5083</td>
<td>$1.0802 \times 10^{-2}$</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1.5000</td>
<td>3.0000</td>
<td>3.0000</td>
<td>4.1533</td>
<td>5.1235</td>
<td>1.5083</td>
<td>$7.1160 \times 10^{-7}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Example 4.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$G$</th>
<th>$|d|_\infty$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.8057</td>
<td>3.1649</td>
<td>1.2158</td>
<td>5.9609</td>
<td>6.8645</td>
<td>61.050</td>
<td>3.9043</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>-1.8728</td>
<td>3.1210</td>
<td>1.1091</td>
<td>5.9941</td>
<td>6.9780</td>
<td>18.293</td>
<td>$1.1353 \times 10^{-1}$</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>-1.8715</td>
<td>3.1180</td>
<td>1.1083</td>
<td>5.9940</td>
<td>6.9776</td>
<td>18.278</td>
<td>$3.0672 \times 10^{-2}$</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>-1.8715</td>
<td>3.1129</td>
<td>1.1083</td>
<td>5.9940</td>
<td>6.9791</td>
<td>18.272</td>
<td>$5.0900 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>-1.8715</td>
<td>3.1129</td>
<td>1.1083</td>
<td>5.9940</td>
<td>6.9791</td>
<td>18.272</td>
<td>$2.1625 \times 10^{-6}$</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>-1.8715</td>
<td>3.1129</td>
<td>1.1083</td>
<td>5.9940</td>
<td>6.9791</td>
<td>18.272</td>
<td>$1.2940 \times 10^{-14}$</td>
<td></td>
</tr>
</tbody>
</table>
6 Conclusion

We have been concerned here with the use of the $l_1$ norm in ODR problem for fitting two concentric circles (or spheres) to data. The Gauss-Newton type method or Levenberg-Marquadt method could be applied for the problem. This problem has been illustrated by the numerical examples, which show that the use of $l_1$ ODR is very rigorous for fitting two concentric circles (or spheres). It is worth noting that the main factor in local convergence is the number of zero distances. Moreover, The number of iterations on fitting two concentric circles (or spheres) by the $l_1$ ODR have been not generally affected by the value of $m$.

Acknowledgments
My most profound gratitude is to Watson, G.A. for the comments he made on the first version of this paper.

References


**Received: October, 2008**