OPERATIONS RESEARCH
Vol. 53, No. 6, November–December 2005, pp. 957–967
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5306 | 0957

DOI 10.1287/opre.1050.0253
© 2005 INFORMS

Majorization Algorithms for Inspecting Circles, Ellipses, Squares, Rectangles, and Rhombi

K. Van Deun
Department of Psychology, Catholic University of Leuven, Tiensestraat 102, B-3000 Leuven, Belgium,
katrijn.vandeun@psy.kuleuven.ac.be

P. J. F. Groenen
Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands,
groenen@few.eur.nl

In several disciplines as diverse as shape analysis, location theory, quality control, archaeology, and psychometrics, it can be of interest to fit a circle through a set of points. We use the result that it suffices to locate a center for which the variance of the distances from the center to a set of given points is minimal. In this paper, we propose a new algorithm based on iterative majorization to locate the center. This algorithm is guaranteed to yield a series of nonincreasing variances until a stationary point is obtained. In all practical cases, the stationary point turns out to be a local minimum. Numerical experiments show that the majorizing algorithm is stable and fast. In addition, we extend the method to fit other shapes, such as a square, an ellipse, a rectangle, and a rhombus by making use of the class of l_p distances and dimension weighting. In addition, we allow for rotations for shapes that might be rotated in the plane. We illustrate how this extended algorithm can be used as a tool for shape recognition.

Subject classifications: mathematics: functions: majorizing functions; facilities: location: continuous; engineering: shape analysis.
Area of review: Optimization.
History: Received September 2003; revision received October 2004; accepted October 2004.

1. Introduction

The central problem of this paper can be described in a simple manner: given a set of points, what is the best circle through these points. This task is equivalent to finding a center point such that its distance to all of the given points is equal. It turns out that this problem appears in several disciplines. For example, in location analysis, this objective is known as the balancing or equity objective (Eiselt and Laporte 1995) and is often applied to decisions where the public needs to be served in a fair manner. An example is finding a location for a new library such that it is located at more or less the same distance for all its users. In the domain of pattern recognition, this criterion is known as least squares orthogonal distance fitting (see, for example, Ahn et al. 2001) with applications in image processing. The problem of finding a circle that comes closest to a set of points forms a special case, with applications, for example, in archeology (Rorres and Romano 1997), computerized roundness inspection (Van-Ban and Lee 1991, Yeralan and Ventura 1988), and unfolding (Van Deun et al. 2005).

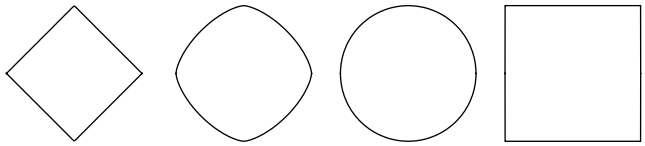
Several algorithms have been proposed that solve the minimum variance problem. Many of these algorithms are computationally slow and strongly depend on the starting position (Ahn et al. 2001). Often they are developed for a specific shape in a certain dimensionality usually chosen to be two or three. In this paper, we will show how the numerical optimization technique of iterative majorization can be used as a robust and computationally inexpensive tool to fit different shapes. Here, we will consider the two-dimensional shapes of circles, ellipses, squares, rectangles, and rhombi that are possibly rotated. Even though this paper is limited to two-dimensional shapes, the proposed algorithms can be used for more than two dimensions as well.

Consider the situation that a circle needs to be fitted through a given set of points where the coordinates of n points are given by the rows of the $n \times 2$ matrix $\mathbf{Z}$ and row i is given by the row vector $\mathbf{z}_i'$. Throughout this paper, we shall denote vectors by boldface lowercase roman letters and matrices by boldface uppercase roman letters. If all $\mathbf{z}_i$ lie exactly on a circle, then there exists a center point $\mathbf{x}$ and a radius R such that all the distances d_i between $\mathbf{x}$ and $\mathbf{z}_i$ equals R. Therefore, the minimization of

$$L(\mathbf{x}, R) = \sum_{i=1}^{n} (d_i - R)^2 \tag{1}$$

has a global minimum of zero whenever the points $\mathbf{z}_i$ fit exactly on a circle. If the points are almost on a circle, then (1) can be used as a least squares model to fit a circle. Note that the size of the circle is not important as it is modelled by the radius R.

Figure 1. Contour line for points with unity distance to the origin.



(a) l_1 ball (b) $l_{3/2}$ ball (c) l_2 ball (d) l_∞ ball

Note. The different panels refer to different l_p metrics.

To fit a circle, d_i needs to be defined as the Euclidean distance. Here, we propose to use its generalization, the Minkowski or l_p-distance

$$d_i = \left(\sum_{s=1}^{m} |u_{is}|^p\right)^{1/p}, \tag{2}$$

with $p \geqslant 1$ the so-called Minkowski parameter, m the dimensionality (throughout this paper $m = 2$), and

$$u_{is} = x_s - z_{is}. \tag{3}$$

Special cases of the l_p-distance are the Euclidean distance for $p = 2$, the rectangular or city-block distance for $p = 1$, and the infinity norm or dominance distance for $p = \infty$. The different shapes that can be fitted by (1) for different choices of p are given in Figure 1. The shape corresponding to the Euclidean distance (l_2) is the well-known circle, for the rectangular distance (l_1) a diamond (in two dimensions a rotated square), and for the dominance distance (l_∞) a square. For example, minimizing (1) for $p = 1$ implies searching for the best-fitting diamond.

Drezner et al. (2002) showed that minimizing (1) is equivalent to minimizing

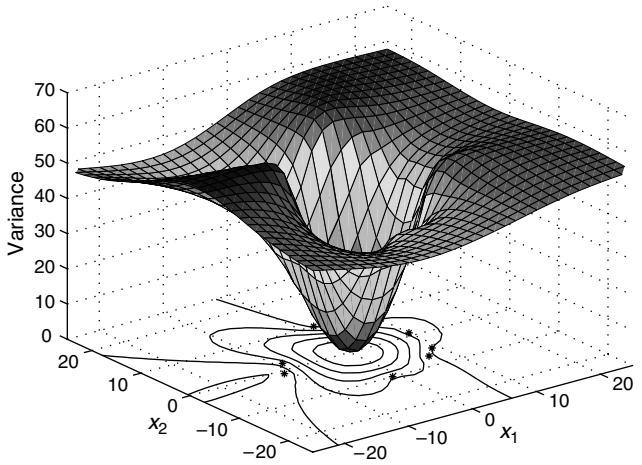
$$\begin{aligned} L(\mathbf{x}) &= \sum_{i=1}^{n}\left(d_i - \frac{1}{n}\sum_i d_i\right)^2 \\ &= \sum_{i=1}^{n} d_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} d_i\right)^2, \end{aligned} \tag{4}$$

which is equal to the variance of the distances if $L(\mathbf{x})$ is divided by n. The advantage of (4) over (1) is that we no longer need to estimate the radius but only the coordinates of the center. In addition, (4) allows us to graphically display $L(\mathbf{x})$ as a function of the two center coordinates. Consider the data reported by Drezner et al. (2002) of nine points that lie approximately on a circle (see Table 1). Figure 2 shows the value of the loss function as a function of the center coordinates $\mathbf{x}$. Note that the nine points are also plotted as stars in the horizontal plane. In Figure 2, it is easy to see that $L(\mathbf{x})$ is not necessarily convex and has a steep decline in the region bounded by the given points.

Table 1. Coordinates for nine points.

	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$	$\mathbf{z}_6$	$\mathbf{z}_7$	$\mathbf{z}_8$	$\mathbf{z}_9$
First dimension	−9	−11	2	−1	4	9	7	7	10
Second dimension	2	−1	10	−10	9	−5	7	−7	1

Figure 2. Variance in function of the center coordinates.



Note. The stars indicate the given points $\mathbf{z}_i$.

This paper is organized as follows. First, we will develop an iterative majorization algorithm that can be used to find the point with minimum variance of the distances to a fixed set of points. Then, the algorithm is applied to the problem of finding the circle that comes closest to a set of points, and we compare it with a Newton-Raphson approach developed by Drezner et al. (2002). Finally, we will show how our algorithm can be extended to model different shapes, such as squares, ellipses, and rectangles. For these shapes, we also allow for rotations. As a possible application, we consider the problem of shape recognition.

2. Minimizing $L(\mathbf{x})$ by Iterative Majorization

To solve the problem of finding coordinates for the center that minimize (1) or, equivalently, (4) an iterative majorization procedure is developed. First, a general description of iterative majorization is presented. Then, we show how a majorizing function that is quadratic in the coordinates can be obtained.

2.1. Iterative Majorization

Iterative majorization is a guaranteed descent method that has been mostly developed in the area of psychometrics and statistics. For a general introduction to iterative majorization, we refer to De Leeuw (1994), Heiser (1995), Lange et al. (2000), and Kiers (2002). Here, we give a short description. The main idea is to replace in each iteration the objective function by an auxiliary function (called the majorizing function) that is easier to minimize, such as a quadratic function. Then, an update of the majorizing function is easy to find. By choosing the majorizing

function in a special manner, the important property of iterative majorization can be proven, that is, the property that the function value will never increase in subsequent iterations. In other words, for almost all practical applications, iterative majorization guarantees that the function values decrease. If the original function is bounded below by zero, as (4), the sequence of function values converges.

Let the majorizing function be denoted by $g(\mathbf{x}, \mathbf{x}_0)$, where $\mathbf{x}_0$ is called the supporting point, which is the current known estimate of $\mathbf{x}$. The majorizing function has to have the following properties:

(1) The majorizing function $g(\mathbf{x}, \mathbf{x}_0)$ should be easier to minimize than the original function $f(\mathbf{x})$.

(2) The majorizing function is larger than or equal to the original function, that is, $f(\mathbf{x}) \leqslant g(\mathbf{x}, \mathbf{x}_0)$ for all $\mathbf{x}$.

(3) The majorizing function touches the original function at the supporting point $\mathbf{x}_0$, that is, $g(\mathbf{x}_0, \mathbf{x}_0) = f(\mathbf{x}_0)$.

Let $\mathbf{x}^+$ be the minimum of $g(\mathbf{x}, \mathbf{x}_0)$. From the two last properties, it follows that

$$f(\mathbf{x}^+) \leqslant g(\mathbf{x}^+, \mathbf{x}_0) \leqslant g(\mathbf{x}_0, \mathbf{x}_0) = f(\mathbf{x}_0), \tag{5}$$

which is called the sandwich inequality by De Leeuw (1994). By choosing $\mathbf{x}_0$ of the current iteration as $\mathbf{x}^+$ from the previous iteration, it follows that a majorizing algorithm yields a nonincreasing sequence of function values. Even though it can only be guaranteed that the algorithm will stop at a stationary point (provided that the function is differentiable), the guaranteed descent property is in almost all practical applications sufficient to stop at a local minimum. In general, iterative majorization methods have linear convergence (De Leeuw 1994), but this slow convergence is compensated for by their computational efficiency. Therefore, in practice with limited precision these methods can have a higher computational speed, than for example, Newton Raphson methods that have a quadratic rate of convergence. On the other hand, the algorithm might be accelerated by gradually shifting from iterative majorization to Newton methods (see Lange et al. 2000 for an example).

Generally, the difficult part of iterative majorization lies in finding the majorizing function. However, one can use the overview of building blocks for majorizing functions given by Kiers (2002) and Groenen (2002). One guideline can be given: as the majorizing function should be easy to minimize, we search for a linear or quadratic majorizer. Concave functions are majorized by a linear function while functions with a bounded Hessian can be majorized by a quadratic function. So, here we will look for a majorizing function that is quadratic in $\mathbf{x}$.

2.2. Obtaining a Majorizing Function for $L(\mathbf{x})$

We now develop a majorizing function for $L(\mathbf{x})$. We use the fact that majorizing functions can be majorized themselves as well, and the algorithm remains convergent. Also, as the majorizing relation is closed under summation, a function consisting of different terms can be majorized by majorizing each of those terms that are not linear or quadratic. Below, we develop the majorizing function in steps. Note that so far $L(\mathbf{x})$ is expressed as a function of distances and that the desired majorizing function has to be a quadratic function in $\mathbf{x}$. Previous work in the domain of multidimensional scaling and summarized by Groenen (2002), found majorizing functions that are linear or quadratic in the coordinates for functions that are either quadratic in d_i or a sum of $-d_i$. Therefore, we take the following steps in majorizing $L(\mathbf{x})$. First, we majorize $L(\mathbf{x})$ to obtain a function that is either quadratic in d_i or is a sum of $-d_i$. Then, we majorize d_i^2 and $-d_i$ by functions that are linear or quadratic in the coordinates. Finally, these results are combined to obtain a quadratic function in $\mathbf{x}$. Although we have taken great care to develop the majorizing algorithm, we cannot exclude the possibility that other majorizing functions exist that might be computationally more efficient.

The first term of $L(\mathbf{x})$ consists of a sum of d_i^2. The second term of $L(\mathbf{x})$, $-n^{-1}(\sum_{i=1}^n d_i)^2$, can be seen as n^{-1} times minus the square of $(\sum_{i=1}^n d_i)$. Developing the inequality $c(t - t_0)^2 \geqslant 0$ gives

$$c(t^2 + t_0^2 - 2tt_0) \geqslant 0,$$
$$-ct^2 \leqslant -2ctt_0 + ct_0^2,$$

where c is a positive value. Substituting $t = \sum_i d_i$ and $t_0 = \sum_i d_{i0}$ gives

$$L(\mathbf{x}) \leqslant \sum_{i=1}^n d_i^2 - \frac{2}{n}\left(\sum_{i=1}^n d_i\right)\left(\sum_{i=1}^n d_{i0}\right) + k_1, \tag{6}$$

with d_{i0} the distances based on $\mathbf{x}_0$ (the previous estimation of $\mathbf{x}$) and $k_1 = n^{-1}(\sum_i d_{i0})^2$. Thus, the right-hand part of (6) gives a majorizing function that consists of squared distances and a weighted sum of minus the distances.

We now focus on majorizing $-d_i$. Making use of Hölder's inequality, Groenen et al. (1999) found a majorizing function for $-d_i$ given as

$$-d_i \leqslant -\sum_{s=1}^m u_{is} u_{is0} b_{is}^{(1)}, \tag{7}$$

where u_{is0} is defined by (3) using the previous estimate $\mathbf{x}_0$ instead of $\mathbf{x}$ and

$$b_{is}^{(1)} = \begin{cases} |u_{is0}|^{p-2}/d_{i0}^{p-1} & \text{if } |u_{is0}| > 0 \text{ and } 1 \leqslant p < \infty, \\ 0 & \text{if } |u_{is0}| = 0 \text{ and } 1 \leqslant p < \infty, \\ |u_{is0}|^{-1} & \text{if } |u_{is0}| > 0, p = \infty, \text{ and } s = s^*, \\ 0 & \text{if } (|u_{is0}| = 0 \text{ or } s \neq s^*) \text{ and } p = \infty, \end{cases} \tag{8}$$

where s^* is defined for $p = \infty$ as the dimension on which the distance is maximal, i.e., $d_i = |u_{is^*}|$.

For d_i^2, Groenen et al. (1999) also gave a quadratic majorizing function by

$$d_i^2 \leqslant \sum_{s=1}^{m} a_{is} u_{is}^2 - 2\sum_{s=1}^{m} u_{is} u_{is0} b_{is}^{(2)} + \sum_{s=1}^{m} c_{is}, \tag{9}$$

where

$$a_{is} = \begin{cases} |u_{is0}|^{p-2}/d_{i0}^{p-2} & \text{if } |u_{is0}| > \epsilon \text{ and } 1 \leqslant p \leqslant 2, \\ |\epsilon|^{p-2}/d_{i0}^{p-2} & \text{if } |u_{is0}| \leqslant \epsilon \text{ and } 1 \leqslant p \leqslant 2, \\ 2(p-1), & 2 < p < \infty, \\ |u_{i\phi_1 0}|/(|u_{i\phi_1 0}| - |u_{i\phi_2 0}|) & \\ \qquad \text{if } |u_{i\phi_1 0}| - |u_{i\phi_2 0}| > \epsilon \text{ and } p = \infty, & \\ (|u_{i\phi_1 0}| + \epsilon)/\epsilon & \text{if } |u_{i\phi_1 0}| - |u_{i\phi_2 0}| \leqslant \epsilon \text{ and } p = \infty, \end{cases} \tag{10}$$

with ϕ_s an index that orders the values of $|u_{is0}|$ decreasingly (thus $|u_{i\phi_1 0}| \geqslant |u_{i\phi_2 0}|$),

$$b_{is}^{(2)} = \begin{cases} 0 & \text{if } 1 \leqslant p \leqslant 2, \\ a_{is} - |u_{is0}|^{p-2}/d_{i0}^{p-2} & \text{if } |u_{is0}| > 0 \text{ and } 2 < p < \infty, \\ 0 & \text{if } |u_{is0}| = 0 \text{ and } 2 < p < \infty, \\ a_{is} & \text{if } s \neq \phi_1 \text{ and } p = \infty, \\ a_{is}|u_{i\phi_2 0}|/|u_{i\phi_1 0}| & \text{if } s = \phi_1,\ p = \infty, \end{cases} \tag{11}$$

and

$$c_{is} = \begin{cases} 0 & \text{if } 1 \leqslant p \leqslant 2, \\ \sum_s a_{is} u_{is0}^2 - d_{i0}^2 & \text{if } 2 < p < \infty, \\ \sum_s \left(2b_{is}^{(2)} - a_{is}\right) u_{is0}^2 + d_{i0}^2 & \text{if } p = \infty. \end{cases} \tag{12}$$

For the proof that (7) and (9) are indeed majorizing functions, we refer to Groenen et al. (1999). Note that if $|u_{is0}| = 0$ for all s, then $d_{i0} = 0$ so that d_{i0}^{p-2} appearing in a_{is} is not defined. In this case, we simply replace $|u_{is0}| = \epsilon$ for some small ϵ so that $d_{i0} > \epsilon$ in a_{is}. Strictly speaking, this adaptation violates the majorization requirements. However, by choosing ϵ sufficiently small the majorizing algorithm remains convergent in practice.

Applying the majorizing inequalities of (7) and (9) into the right-hand part of (6) gives

$$\begin{aligned} L(\mathbf{x}) &\leqslant \sum_{i=1}^{n} d_i^2 - \frac{2}{n}\left(\sum_{i=1}^{n} d_i\right)\left(\sum_{i=1}^{n} d_{i0}\right) + k_1 \\ &\leqslant \sum_{is} a_{is} u_{is}^2 - 2\sum_{is} u_{is} b_{is} + k_2, \end{aligned} \tag{13}$$

where $b_{is} = [(n^{-1}\sum_i d_{i0}) b_{is}^{(1)} + b_{is}^{(2)}] u_{is0}$ and $k_2 = k_1 + \sum_{is} c_{is}$. Inserting $x_s - z_{is}$ for u_{is} in the right-hand part of (13) gives

$$\begin{aligned} L(\mathbf{x}) &\leqslant g_1(\mathbf{x}, \mathbf{x}_0) \\ &= \sum_{is} a_{is}(x_s - z_{is})^2 - 2\sum_{is}(x_s - z_{is}) b_{is} + k_2 \\ &= \sum_s x_s^2 \sum_i a_{is} - 2\sum_s x_s \sum_i (a_{is} z_{is} + b_{is}) + k_3, \end{aligned} \tag{14}$$

where $k_3 = k_2 + \sum_{is} a_{is} z_{is}^2 + 2\sum_{is} z_{is} b_{is}$. Inspecting $g_1(\mathbf{x}, \mathbf{x}_0)$ shows that the majorizing function in the right-hand part of (14) is indeed quadratic in x_s. The minimum of a quadratic function can easily be obtained by setting the derivative equal to zero and solving for x_s, giving the update

$$x_s^+ = \frac{\sum_i (a_{is} z_{is} + b_{is})}{\sum_i a_{is}}. \tag{15}$$

The iterative majorization algorithm is given by

Step 1. Choose initial $\mathbf{x}^{(0)}$. Set iteration counter $l := 0$ and set the convergence criterion ϵ_{conv} to a small positive constant.

Step 2. $l := l + 1$.

Step 3. Calculate the update $\mathbf{x}^+$ using (15), where $\mathbf{x}_0 = \mathbf{x}^{(l-1)}$.

Step 4. $\mathbf{x}^{(l+1)} = 2\mathbf{x}^+ - \mathbf{x}^{(l-1)}$.

Step 5. Check for convergence: If $\|\mathbf{x}^{(l-1)} - \mathbf{x}^{(l)}\| > \epsilon_{\text{conv}}$ go to Step 2.

As a rational starting point for $\mathbf{x}^{(0)}$, we use the centroid of the data points, i.e., $\mathbf{x}^{(0)} = n^{-1}\sum_i \mathbf{z}_i$.

Step 4 is an acceleration method that can be used with iterative majorization (De Leeuw and Heiser 1980, Heiser 1995). It can be proved that the use of the so-called relaxed update still guarantees a nonincreasing sequence of the variance of the distances (see Groenen et al. 1995), while experience shows that the number of iterations will be approximately halved.

2.3. Minimizing the Variance of Euclidean Distances by Iterative Majorization

For fitting circles, we have to use Euclidean distances. In this case, the update (15) of the majorizing algorithm simplifies considerably. In fact, for $p = 2$,

$$\begin{aligned} a_{is} &= 1, \\ b_{is}^{(1)} &= \begin{cases} d_{i0}^{-1} & \text{if } d_{i0} > 0, \\ 0 & \text{if } d_{i0} = 0, \end{cases} \\ b_{is}^{(2)} &= 0, \\ c_{is} &= 0, \\ b_{is} &= \left(n^{-1}\sum_i d_{i0}\right) b_{is}^{(1)} u_{is0}, \end{aligned}$$

yielding the update

$$x_s^+ = n^{-1} \sum_i (z_{is} + b_{is}). \tag{16}$$

For Euclidean distances, the majorizing function is illustrated in Figure 3, where both the original loss function $L(\mathbf{x})$ and the majorizing function from the right-hand side of (14) are plotted for the small data example of Table 1. Here, the supporting point is chosen as $\mathbf{x}_0' = [2\ 0]$. As can be seen in Figure 3a, the majorizing function is a quadratic function that lies above the original loss function. In Figure 3b, we see that the majorizing function indeed touches the original function at the supporting point $\mathbf{x}_0$.

3. Application: Finding the Best-Fitting Circle

We illustrate our algorithm by applying it to the problem of finding a circle that fits some given points. Several solutions have been proposed that minimize the variance of the distances of the center to the given points (see, for example, Ahn et al. 2001, Drezner et al. 2002, Rorres and Romano 1997). Here, we limit our comparison to the iterative majorization algorithm and the Newton-Raphson algorithm proposed by Drezner et al. (2002). At the end of this section, we will further investigate the performance of the iterative majorization approach in a simulation study.

First, we compare the sensitivity of the two algorithms for the starting values of $\mathbf{x}^{(0)}$. Again, we use the nine points reported in Table 1. Both algorithms were stopped when the difference between the $L(\mathbf{x})$ values in two subsequent iterations was less than 10^{-5}. This weaker convergence criterion was chosen for reasons given below. To check the sensitivity for the start configuration, we started both algorithms from all points on a grid defined for x_1 and x_2 between $[-20, 20]$ in steps of 2. Then, the final $L(\mathbf{x})$ after convergence was noted. Figure 4a shows these values on the z-axis for the Newton-Raphson algorithm of Drezner et al. (2002) and for the majorization algorithm in Figure 4b.

Figure 3. Plot of the original loss function and the quadratic majorizing function with supporting point $\mathbf{x}_0' = [2\ 0]$ (Panel a). Panel (b) shows a detail.

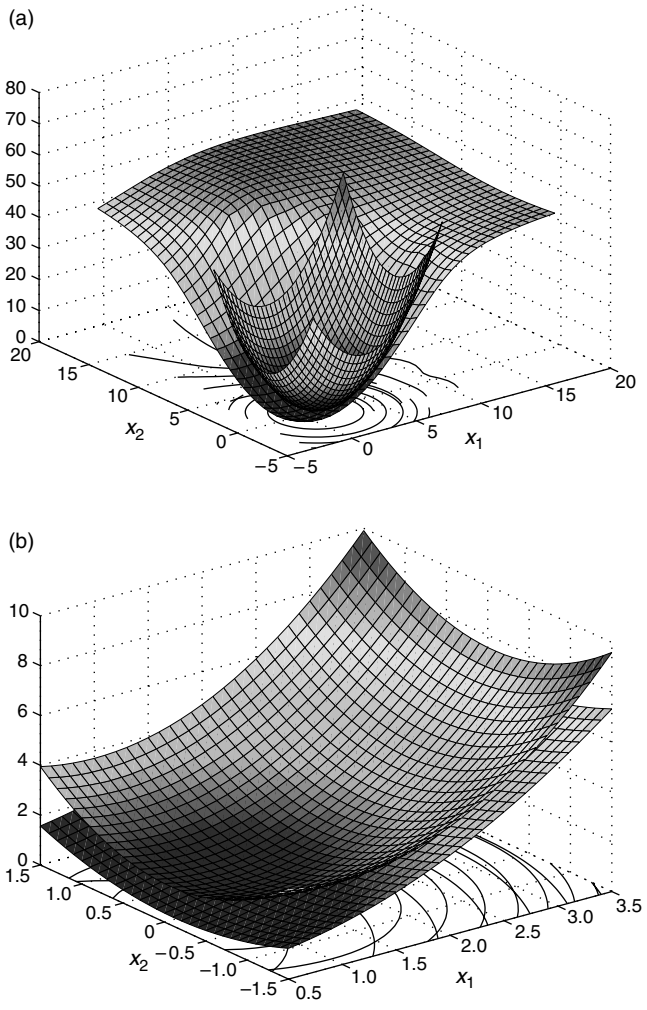
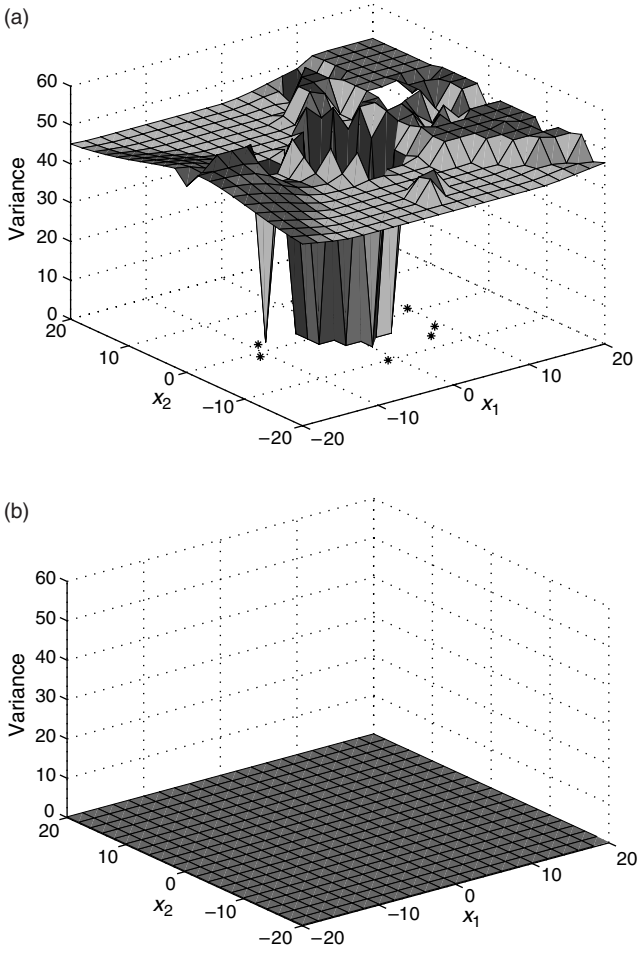


Figure 4. Value of the objective function for different starting positions.



Note. The upper panel stems from the Newton-Raphson approach and the bottom panel from the iterative majorization approach. The stars indicate the given points.

Table 2. Some measures of performance of the algorithms for the Drezner data. The first line shows the results for the Newton-Raphson approach; the second line for the iterative majorization algorithm.

Algorithm	Min fit	Range fit	Min iter	Max iter	Time
Newton-Raphson	0.19883320109211	9.5e−16	3	4	19.92
Iterative majorization	0.19883320109211	6.4e−13	4	7	20.12

It can be seen clearly that the Newton-Raphson algorithm strongly depends on the starting position. Only for starting positions not too far from the center, the global minimum of $L(\mathbf{x})$ is found. Note that for some starting values, there is a gap in the figure. This gap is due to numerical problems that turn out to be more severe for stronger convergence criteria. The majorizing algorithm shows a flat surface indicating that from all starting positions with $-20 \leqslant x_1 \leqslant 20$ and $-20 \leqslant x_2 \leqslant 20$, the global minimum of $L(\mathbf{x})$ is found. So, even starting outside the circle leads to the global optimum.

Next, we compare both algorithms for starting positions close to the optimum. As a convergence criterion we take the distance between the updated center and the previous one: when this distance is smaller than 10^{-5}, the iterative procedure is stopped. One thousand starting positions were generated uniformly on the disk centered around the optimum and with radius equal to one. The results are reported in Table 2. For each algorithm the minimum value of $L(\mathbf{x})$, the range of the $L(\mathbf{x})$ values, the minimum number of iterations, the maximum number of iterations, and the total computing time are given. In terms of convergence, the Newton-Raphson approach is faster as can be seen from the number of iterations and the range of the variance. However, in terms of CPU time the iterative majorization algorithm is about as fast.

We further investigate the performance of the iterative majorization approach in a small simulation study. In this study three factors are varied:

- the number of given points ($n = 3, 4, 6, 10, 20$),
- the noise level ($e =$ no noise, 5%), and
- the angle the given points cover on the circle ($a =$ 90, 180, and 360 degrees).

Per case, 100 times n points were generated on a part (covering a degrees) of the circle with origin $(0, 0)$ and radius one plus e. For each case, a starting point $\mathbf{x}^{(0)}$ was generated uniformly on the disc with radius one. The algorithms terminate using $\epsilon_{\text{conv}} = 10^{-5}$. In this study, the true center location was known a priori. Therefore, the success of the algorithm can be measured by the Euclidean distance between the center location found by the algorithm and the true origin at $(0, 0)$. Figures 5 and 6 show the distribution of these distances by boxplots for the various combinations of the factors in this study. Note that because the distributions are highly skew, a log scale is used in these boxplots.

To investigate what effects are important we did an analysis of variance (ANOVA) using the log of the distances as the response variable; see Table 3. The number of given points n is treated as a random factor, while both the coverage angle a and the error level e are treated as fixed factors. The effect size reflects the proportion of variance of the log distances that is accounted for by the effect. In total, 86% is accounted for. The main effects account for almost all explained variance, with 52% accounted for by the error level.

Figure 5. Boxplots of the distances between the true center and the center located by the algorithm in the simulation study for different n and a for error free data.

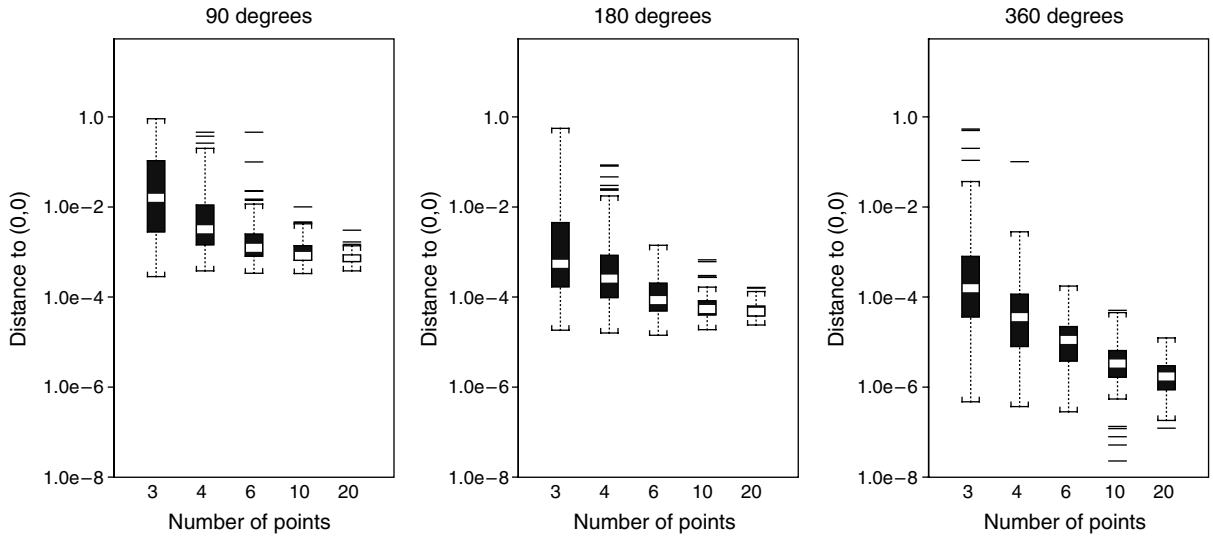
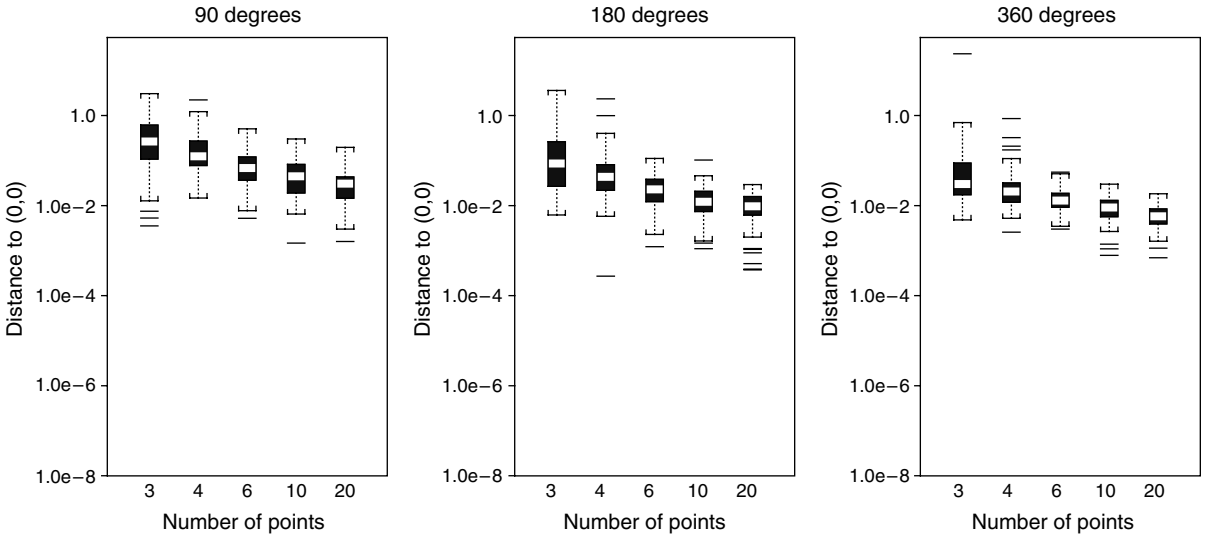


Figure 6. Boxplots of the distances between the true center and the center located by the algorithm in the simulation study for different n and a for error perturbed data.



There is a large difference between the case with and without error: in absence of error, the solution lies clearly closer to the true origin. For the error-free case (see Figure 5), both the angle and the number of points are influential. The solutions are clearly closer to the true solutions for a larger number of points and for larger coverage angles. This is less so for the case with error (Figure 6).

4. Extensions of Majorization

So far we have seen how the best-fitting circle through a number of given points can be found. The loss function we used is based on l_p distances, and we derived a majorizing function that is easy to minimize for l_p distances with $p \geqslant 1$. This majorizing function was used to solve the minimum variance problem for Euclidean distances with, as an application, the problem of finding a circle. Minimizing the variance for other l_p norms allows us to fit different shapes such as the ones given in the first row of Figure 7. For example, taking l_1 or l_∞ distances can be used to find the best fitting diamond or square, respectively.

Table 3. Analysis of variance table for log distances in the simulation study.

Source	SS	DF	MS	F	Effect size
e	3,712	1	3,712	330.76	0.52
a	1,156	2	578	558.79	0.16
n	654	4	163	502.00	0.11
$e \times a$	311	2	155	68.63	0.04
$e \times n$	45	4	11	34.43	0.02
$a \times n$	8	8	1	3.17	0
$e \times a \times n$	18	8	2	6.96	0
Residual	968	2,970	0.33		
Total	6,873	2,999			

The loss function can be extended to include dimension weights and rotations. Dimension weights are used to differentially stretch the dimensions so that instead of a circle, diamond, or square we can fit an ellipse, rhombus, or rectangle (see the second row of Figure 7). The effect of rotations is illustrated in the third row of the same figure, while the combined effect of dimension weighting and rotating results in the shapes in the bottom row of Figure 7. Note that the order of the transformations has to be respected; first apply dimension weights to the basic shapes of the top row in Figure 7 and then rotate to obtain the shapes in the bottom row.

In the remainder of this section, we will first introduce rotations and dimension weighting formally. Then, we will illustrate how the algorithm can be used to solve the problem of shape recognition for a restricted class of shapes. An extension of the majorizing algorithm that searches for optimal center coordinates, dimension weights, and rotation is developed in the appendix.

4.1. Rotations and Dimension Weighting

Let $\mathbf{T}$ be a rotation matrix. Note that for any rotation matrix $\mathbf{T}$ is orthonormal, that is, $\mathbf{T}'\mathbf{T} = \mathbf{T}^{-1}\mathbf{T} = \mathbf{I}$. In two dimensions, $\mathbf{T}$ can be written as

$$\mathbf{T} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \tag{17}$$

where α is the rotation angle anticlockwise. Thus, $\mathbf{ZT}$ is an anticlockwise rotation of $\mathbf{Z}$ by α degrees. Choosing $\mathbf{T}$ as the identity matrix $\mathbf{I}$ leaves $\mathbf{ZT} = \mathbf{ZI} = \mathbf{Z}$ unrotated. We also consider dimension weighting by a diagonal matrix $\mathbf{W}$, so that, in $\mathbf{ZW}$ the coordinates of $\mathbf{Z}$ along dimension s are weighted by the factor w_s. Rotations and weighting can be

Figure 7. Some shapes that can be fitted with l_p distances for $p \geqslant 1$, allowing for dimension weighting and orthogonal rotations.

Rotation	Dimension weighting	$p = 1.0$	$p = 1.5$	$p = 2.0$	$p = 3.0$	$p = \infty$
No	No					
No	Yes					
Yes	No					
Yes	Yes					

Note. Each column illustrates a certain l_p norm. The rows show the shape that can be fitted, whether it is weighted (by $w_1 = 1$ and $w_2 = 0.3$) or unweighted ($w_1 = 1$ and $w_2 = 1$), and rotated by 30 degrees or not.

combined by **ZTW**, which simply means first rotate **Z** by **T** and then weight the rotated axes by **W**.

To adapt the variance function for different rotations and dimension weighting, we have to minimize

$$L_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) = \sum_{i=1}^{n} d_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} d_i\right)^2 \tag{18}$$

with u_{is} in (2) defined as

$$u_{is} = x_s - w_s \mathbf{z}_i' \mathbf{t}_s, \tag{19}$$

where $\mathbf{t}_s$ is column s of the rotation matrix **T**. Here, the right order of operations is first to rotate and then weight the data. Only in this way we obtain an l_p ball with minimum variance of the distances to the center. Some precaution is necessary, because minimizing $L_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$ will trivially be zero when choosing $\mathbf{x} = \mathbf{0}$ and $\mathbf{W} = \mathbf{0}$. Therefore, we impose the additional restriction that $\operatorname{tr}\mathbf{W}^2 = \sum_s w_s^2 = m$ to avoid the degeneracy. As preliminary experimentation showed that occasionally solutions occurred with only one nonzero weight, we further impose the restriction that $w_s \geqslant w_{\min}$ for all s. $w_{\min}$ is a small positive constant. In the appendix we develop an iterative majorization algorithm that can be used to find **x**, **T**, and **W** such that L_2 is minimized.

4.2. Shape Recognition

So far we considered the problem of fitting a predefined shape, but the algorithm could also be used for shape recognition. For example, in reverse engineering (with applications in reproduction and redesign) the goal is to model engineering parts according to the original design (see, for example, Langbein et al. 2001), while in graphics recognition (see, for example, Fonseca and Jorge 2001) one wants to digitally recognize hand-drawn shapes. Apart from the problem that the model has to be noiseless, the shape has to be recognized under rotations and unknown scale. In the appendix, we developed an algorithm that can be used to solve this problem for a restricted class of shapes. Although we can fit any of the shapes in Figure 7 to our data (and all other l_p balls with $p \geqslant 1$), we limit it to the case of the last row in Figure 7 (thus allowing for rotation and dimension weighting) for $p = 1$, 2, and ∞.

The proposed algorithm can be used to fit a weighted and rotated l_p ball to the given points. By doing this for different p, the shape with the highest fit can be found. It is this shape that we will retain as the origin of the points. As a measure of fit we will consider a normalized variance. This is the variance of cd_i, where c is chosen such that the mean distance (and thus the radius R) equals one. In this way, the variances are comparable between different data sets. We give a small illustration for noiseless data.

Twelve points were generated on each of three different shapes: a diamond, an unrotated ellipsis (with weights $w_1 = 1.35$ and $w_2 = 0.41$), and a rotated ($\alpha = 30$ degrees anticlockwise) rectangle. We use the majorization algorithm given in the appendix to find the center, weights, and rotation angle of the ball. To avoid local minima, 200 solutions were generated on the basis of random initial parameters. The numbers shown in Figure 8 are based on the solution with the lowest variance. For each shape, three rows are given with the estimated weights, rotation angle, and the scaled variance for one of the following norms: l_1, l_2, or l_∞. The shapes are well detected as the original norm is recovered and as the parameter values approximate the underlying ones well. For example, in case of the ellipsis we find the lowest variance for the Euclidean norm, the rotation angle equals zero and the weights equal [1.35 0.41]. In this case, the true underlying values are fully recovered. In case

Figure 8. Shape recognition.

Shape	Norm	α	w′	Scaled variance
	l_1	16	[1.00 1.00]	1.0×10^{-6}
	l_2	58	[1.05 0.95]	1.5×10^{-2}
	l_∞	41	[0.98 1.02]	7.3×10^{-4}
	l_1	164	[1.37 0.36]	2.9×10^{-3}
	l_2	0	[1.35 0.41]	9.9×10^{-8}
	l_∞	121	[1.38 0.32]	7.6×10^{-3}
	l_1	32	[1.40 0.22]	4.0×10^{-2}
	l_2	29	[1.37 0.37]	1.4×10^{-2}
	l_∞	30	[1.37 0.41]	3.6×10^{-5}

Notes. For each shape at the left, the rotation angle, the weights, and the scaled variance obtained with the majorizing algorithm are reported. This is done for the l_1, l_2, and l_∞ norm.

of the diamond, the solution based on dominance distances fits well too, with a rotation angle of almost 45 degrees. This comes as no surprise, given that the diamond is a rotated (by 45 degrees) square.

5. Conclusion and Discussion

We showed how iterative majorization can be used to minimize the variance of distances, which is a popular objective function in location theory and in pattern recognition. To be able to fit both circular and square shapes, we extended the use of the Euclidean distance to l_p distances including the city-block distance and dominance distance. We also showed how spherical, rectangular, and rhombic shapes could be fitted by including dimension weights in the loss function. In this case, too, we used iterative majorization to develop an algorithm to minimize the variance of the distances. A final extension is to include rotations so that the shapes may be rotated in space.

In a small numerical application on fitting circles, we showed that our algorithm is equally fast but more robust in comparison to a Newton-Raphson approach that was specifically developed to obtain a fast algorithm. In a different numerical experiment, we illustrated how our proposed algorithm can be used as a tool for shape recognition.

Here, we restricted ourselves to two-dimensional problems, but the algorithms can be applied to higher dimensional problems without modification. In applications where some points may be measured (wrongly) to be far off the shape, it may be wise to robustify the variance function by changing to the mean absolute deviation as a loss criterion. In this case, a majorizing algorithm is possible as well, but it will be necessary to investigate how well such an algorithm is prone to local minima.

The algorithms in this paper can be applied to any number of given points $n > 2$. It remains to be investigated how many points are mathematically necessary to detect a certain shape and how many points are needed in practice to guarantee a proper fit. These issues are to be studied in future research.

Appendix. A Majorizing Algorithm for $L_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$

The majorization derivations remain the same up to (13). Substituting the definition of u_{is} from (19) into (13) gives

$$\begin{aligned} L_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &\leqslant g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) \\ &= \sum_{is} a_{is}(x_s - w_s \mathbf{z}_i' \mathbf{t}_s)^2 - 2\sum_{is}(x_s - w_s \mathbf{z}_i' \mathbf{t}_s) b_{is} + k_2. \end{aligned} \quad (20)$$

Note that the majorizing function $g_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$ is quadratic in $\mathbf{x}$, $\mathbf{T}$, and $\mathbf{W}$. We shall use alternating least squares to update the parameters, that is, $\mathbf{x}$ is updated while $\mathbf{T}$ and $\mathbf{W}$ are kept fixed, then $\mathbf{T}$ is updated while $\mathbf{x}$ and $\mathbf{W}$ are kept fixed, and finally $\mathbf{W}$ is updated while $\mathbf{x}$ and $\mathbf{T}$ are kept fixed.

Updating $\mathbf{x}$ is analogous to (15). $g_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$ can be written as

$$\begin{aligned} g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &= \sum_s x_s^2 \sum_i a_{is} \\ &\quad - 2\sum_s x_s \sum_i (a_{is} w_s \mathbf{z}_i' \mathbf{t}_s + b_{is}) + k_3, \end{aligned} \quad (21)$$

where k_3 is a constant not depending on $\mathbf{x}$, i.e., $k_3 = \sum_{is} a_{is} w_s^2 (\mathbf{z}_i' \mathbf{t}_s)^2 + 2\sum_{is} w_s b_{is} \mathbf{z}_i' \mathbf{t}_s$. It can be shown that the optimal update minimizing $g_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$ over $\mathbf{x}$ while keeping $\mathbf{T}$ and $\mathbf{W}$ fixed equals

$$x_s^+ = \frac{\sum_i (a_{is} w_s \mathbf{z}_i' \mathbf{t}_s + b_{is})}{\sum_i a_{is}}. \quad (22)$$

To update $\mathbf{W}$ for fixed $\mathbf{T}$ and $\mathbf{x}$, we write $g_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$ as

$$\begin{aligned} g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &= \sum_{is} a_{is} w_s^2 (\mathbf{z}_i' \mathbf{t}_s)^2 \\ &\quad - 2\sum_{is} w_s \mathbf{z}_i' \mathbf{t}_s (x_s a_{is} - b_{is}) + k_4, \end{aligned} \quad (23)$$

where k_4 is a constant not depending on $\mathbf{W}$, i.e., $k_4 = \sum_{is} x_s^2 a_{is} - 2\sum_{is} x_s b_{is}$. Note that $\mathbf{W}$ has a length constraint such that $\sum_s w_s^2 = m$. We first rewrite (23) as

$$\begin{aligned} g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &= \sum_s w_s^2 \sum_i [a_{is} (\mathbf{z}_i' \mathbf{t}_s)^2] \\ &\quad - 2\sum_s w_s \left[\sum_i \mathbf{z}_i' \mathbf{t}_s (x_s a_{is} - b_{is}) \right] + k_4 \\ &= \sum_s w_s^2 c_s - 2\sum_s w_s r_s + k_5 \\ &= \mathbf{w}' \mathbf{C}_\mathbf{w} \mathbf{w} - 2\mathbf{w}' \mathbf{r}_\mathbf{w} + k_5, \end{aligned} \quad (24)$$

where $\mathbf{w}$ is a vector with elements w_s, $\mathbf{C}_\mathbf{w}$ is a diagonal matrix with diagonal elements $c_s = a_{is} (\mathbf{z}_i' \mathbf{t}_s)^2$, and $\mathbf{r}_\mathbf{w}$ is a vector with elements $\sum_i \mathbf{z}_i' \mathbf{t}_s (x_s a_{is} - b_{is})$. Because $\mathbf{C}_\mathbf{w}$ is generally not equal to $\mathbf{I}$, the update $\mathbf{C}_\mathbf{w}^{-1} \mathbf{r}_\mathbf{w} / \| \mathbf{C}_\mathbf{w}^{-1} \mathbf{r}_\mathbf{w} \|$ is not the least squares update, because the metric $\mathbf{C}_\mathbf{w}$ in (24) is

not equal to the metric used for the length constraint. We use an additional majorizing step to obtain a majorizing function that has $\mathbf{I}$ as a metric. Let $\lambda_{\mathbf{w}}$ be the largest eigenvalue of $\mathbf{C}_{\mathbf{w}}$. Then, the matrix $\mathbf{C}_{\mathbf{w}} - \lambda_{\mathbf{w}}\mathbf{I}$ is negative semidefinite, so that we obtain the inequality $(\mathbf{w} - \mathbf{w}_0)' \cdot (\mathbf{C}_{\mathbf{w}} - \lambda_{\mathbf{w}}\mathbf{I})(\mathbf{w} - \mathbf{w}_0) \leqslant 0$ with a strict equality if $\mathbf{w} = \mathbf{w}_0$. Expanding this inequality yields

$$\begin{aligned}(\mathbf{w} - \mathbf{w}_0)'\mathbf{C}_{\mathbf{w}}(\mathbf{w} - \mathbf{w}_0) &\leqslant \lambda_{\mathbf{w}}(\mathbf{w} - \mathbf{w}_0)'(\mathbf{w} - \mathbf{w}_0),\\ \mathbf{w}'\mathbf{C}_{\mathbf{w}}\mathbf{w} &\leqslant \lambda_{\mathbf{w}}\mathbf{w}'\mathbf{w} - 2\lambda_{\mathbf{w}}\mathbf{w}'(\mathbf{w}_0 - \lambda_{\mathbf{w}}^{-1}\mathbf{C}\mathbf{w}_0)\\ &\quad + \mathbf{w}_0'(\lambda_{\mathbf{w}}\mathbf{I} - \mathbf{C}_{\mathbf{w}})\mathbf{w}_0,\end{aligned} \tag{25}$$

so that

$$\begin{aligned}g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &\leqslant \lambda_{\mathbf{w}}\mathbf{w}'\mathbf{w} - 2\lambda_{\mathbf{w}}\mathbf{w}'(\mathbf{w}_0 + \lambda_{\mathbf{w}}^{-1}[\mathbf{r}_{\mathbf{w}} - \mathbf{C}_{\mathbf{w}}\mathbf{w}_0])\\ &\quad + \mathbf{w}_0'(\lambda_{\mathbf{w}}\mathbf{I} - \mathbf{C}_{\mathbf{w}})\mathbf{w}_0 + k_4.\end{aligned} \tag{26}$$

In the right-hand part of (26), the metric of the quadratic part in $\mathbf{w}$ does coincide with the metric of the length constraint, so that the minimum of (26) subject to the length constraint is given by

$$\mathbf{w}_u = m^{1/2}\mathbf{q}/\|\mathbf{q}\|, \tag{27}$$

where $\mathbf{q} = \mathbf{w}_0 + \lambda_{\mathbf{w}}^{-1}[\mathbf{r}_{\mathbf{w}} - \mathbf{C}_{\mathbf{w}}\mathbf{w}_0]$. To impose the additional restriction that $w_s \geqslant w_{\min}$, the calculation of the weights should be adapted in the following way for the case that $m = 2$:

$$\begin{cases}\mathbf{w}^+ = \mathbf{w}_u & \text{if } w_{\phi_2} \geqslant w_{\min},\\ w_{\phi_2}^+ = w_{\min} \text{ and } w_{\phi_1}^+ = [m - (w_{\min})^2]^{1/2} & \text{if } w_{\phi_2} < w_{\min},\end{cases} \tag{28}$$

with ϕ_s an index that orders the values of $\mathbf{w}_u$ decreasingly. Note that $w_{\min}$ must be (much) smaller than 1 for a solution to exist. If $m > 2$, a different procedure is needed using an adaptation of the explicit solutions for length-constrained nonnegative least squares discussed in Groenen et al. (2000).

Next, we wish to obtain an update for $\mathbf{T}$ for fixed $\mathbf{x}$ and $\mathbf{W}$. Note that $\mathbf{T}$ is constrained to be orthonormal. Within the framework of iterative majorization, Kiers and Groenen (1996) showed that it is helpful to majorize quadratic functions in $\mathbf{T}$ in such a way that the metric is removed. This can be done by applying again the majorizing inequality of (25). Before we can do so, we rewrite (23) as

$$\begin{aligned}g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &= \sum_s \mathbf{t}_s'\left[w_s^2\sum_i a_{is}\mathbf{z}_i\mathbf{z}_i'\right]\mathbf{t}_s\\ &\quad - 2\sum_s \mathbf{t}_s'\left[w_s\sum_i \mathbf{z}_i(x_s a_{is} - b_{is})\right] + k_4\\ &= \sum_s \mathbf{t}_s'\mathbf{C}_{\mathbf{t}_s}\mathbf{t}_s - 2\sum_s \mathbf{t}_s'\mathbf{r}_{\mathbf{t}_s} + k_4,\end{aligned} \tag{29}$$

where $\mathbf{C}_{\mathbf{t}_s} = w_s^2\sum_i a_{is}\mathbf{z}_i\mathbf{z}_i'$ and $\mathbf{r}_{\mathbf{t}_s} = w_s\sum_i \mathbf{z}_i(x_s a_{is} - b_{is})$. Applying (25) to (29) gives

$$\begin{aligned}g_2(\mathbf{x}, \mathbf{T}, \mathbf{W}) &\leqslant \sum_s \lambda_{\mathbf{t}_s}\mathbf{t}_s'\mathbf{t}_s - 2\sum_s \lambda_{\mathbf{t}_s}\mathbf{t}_s'(\mathbf{t}_{s0} + \lambda_{\mathbf{t}_s}^{-1}[\mathbf{r}_{\mathbf{t}_s} - \mathbf{C}_{\mathbf{t}_s}\mathbf{t}_{s0}])\\ &\quad + \sum_s \mathbf{t}_{s0}'(\lambda_{\mathbf{t}_s}\mathbf{I} - \mathbf{C}_{\mathbf{t}_s})\mathbf{t}_{s0} + k_4\\ &= \sum_s \lambda_{\mathbf{t}_s} - 2\sum_s \mathbf{t}_s'\mathbf{p}_s + k_6\\ &= \sum_s \lambda_{\mathbf{t}_s} - 2\operatorname{tr}\mathbf{T}'\mathbf{P} + k_5,\end{aligned} \tag{30}$$

where $\lambda_{\mathbf{t}_s}$ is the largest eigenvalue of $\mathbf{C}_{\mathbf{t}_s}$, $\mathbf{p}_s = \lambda_{\mathbf{t}_s}(\mathbf{t}_{s0} + \lambda_{\mathbf{t}_s}^{-1}[\mathbf{r}_{\mathbf{t}_s} - \mathbf{C}_{\mathbf{t}_s}\mathbf{t}_{s0}])$, $\mathbf{P}$ has columns $\mathbf{p}_s$, and $k_5 = k_4 + \sum_s \mathbf{t}_{s0}'(\lambda_{\mathbf{t}_s}\mathbf{I} - \mathbf{C}_{\mathbf{t}_s})\mathbf{t}_{s0}$. Note that the simplification $\lambda_{\mathbf{t}_s}\mathbf{t}_s'\mathbf{t}_s = \lambda_{\mathbf{t}_s}$ makes use of the property of rotation matrices that the sum of squares of a column (or a row) equals one. Thus, (30) shows that $g_2(\mathbf{x}, \mathbf{T}, \mathbf{W})$ can be majorized by the linear function in $\mathbf{T}$, $-2\operatorname{tr}\mathbf{T}'\mathbf{P}$ plus a constant. Let $\mathbf{P} = \mathbf{K}\Phi\mathbf{L}'$ be the singular value decomposition with $\mathbf{K}$ and $\mathbf{L}$ orthonormal and Φ diagonal with nonnegative diagonal elements. Then, the minimum of $-2\operatorname{tr}\mathbf{T}'\mathbf{P}$ is obtained for

$$\mathbf{T}^+ = \mathbf{K}\mathbf{L}' \tag{31}$$

(see, for example, Kiers and Groenen 1996).

This leads to the following algorithm. If dimension weighting is necessary, let $\delta_{\mathbf{W}} := 1$, otherwise set $\delta_{\mathbf{W}} := 0$. Similarly, set $\delta_{\mathbf{T}} := 1$ if rotations are necessary, otherwise set $\delta_{\mathbf{T}} := 0$.

Step 1. Choose an initial $\mathbf{x}^{(0)}$. Set iteration counter $l := 0$ and set the convergence criterion ϵ_{conv} to a small positive constant. Set $\mathbf{W}^{(0)} = m^{1/2}\mathbf{I}$ and $\mathbf{T}^{(0)} = \mathbf{I}$.

Step 2. $l := l + 1$.

Step 3. Calculate the update $\mathbf{x}^+$ using formula (22), where $\mathbf{x}_0 = \mathbf{x}^{(l-1)}$, $\mathbf{W} = \mathbf{W}^{(l-1)}$, and $\mathbf{T} = \mathbf{T}^{(l-1)}$. Set $\mathbf{x}^{(l)} := \mathbf{x}^+$.

Step 4. If $\delta_{\mathbf{W}} = 1$, calculate the update $\mathbf{W}^+$ using formula (28) where $\mathbf{W}_0 = \mathbf{W}^{(l-1)}$, $\mathbf{x} = \mathbf{x}^{(l)}$, and $\mathbf{T} = \mathbf{T}^{(l-1)}$. Set $\mathbf{W}^{(l)} := \mathbf{W}^+$.

Step 5. If $\delta_{\mathbf{T}} = 1$, calculate the update $\mathbf{T}^+$ using formula (31), where $\mathbf{T}_0 = \mathbf{T}^{(l-1)}$, $\mathbf{x} = \mathbf{x}^{(l)}$, and $\mathbf{W} = \mathbf{W}^{(l)}$. Set $\mathbf{T}^{(l)} := \mathbf{T}^+$.

Step 6. Check for convergence: If $\|\mathbf{x}^{(l-1)} - \mathbf{x}^{(l)}\| > \epsilon_{\text{conv}}$, go to Step 2.

References

Ahn, S. J., W. Rauh, H.-J. Warnecke. 2001. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition* **34** 2283–2303.

De Leeuw, J. 1994. Block relaxation algorithms in statistics. H. H. Bock, W. Lenski, M. M. Richter, eds. *Information Systems and Data Analysis*. Springer-Verlag, Berlin, Germany, 308–325.

De Leeuw, J., W. J. Heiser. 1980. Multidimensional scaling with restrictions on the configuration. P. R. Krishnaiah, ed. *Multivariate Analysis V*. North-Holland, Amsterdam, The Netherlands, 501–522.

Drezner, Z., S. Steiner, G. O. Wesolowsky. 2002. On the circle closest to a set of points. *Comput. Oper. Res.* **29** 637–650.

Eiselt, H. A., G. Laporte. 1995. Objectives in location problems. Z. Drezner, ed. *Facility Location: A Survey of Applications and Methods*. Springer, New York, 151–180.

Fonseca, M. J., J. A. Jorge. 2001. Experimental evaluation of an online scribble recognizer. *Pattern Recognition Lett.* **22** 1311–1319.

Groenen, P. J. F. 2002. Iterative majorization algorithms for minimizing loss functions in classification. Working paper, *8th Conf. IFCS*. Krakow, Poland.

Groenen, P. J. F., W. J. Heiser, J. J. Meulman. 1999. Global optimization in least-squares multidimensional scaling by distance smoothing. *J. Classification* **16** 225–254.

Groenen, P. J. F., R. Mathar, W. J. Heiser. 1995. The majorization approach to multidimensional scaling for Minkowski distances. *J. Classification* **12** 3–19.

Groenen, P. J. F., B. J. van Os, J. J. Meulman. 2000. Optimal scaling by alternating length-constrained nonnegative least squares, with application to distance-based analysis. *Psychometrika* **65** 511–524.

Heiser, W. J. 1995. Convergent computation by iterative majorization: Theory and applications in multidimensional data analysis. W. J. Krzanowski, ed. *Recent Advances in Descriptive Multivariate Analysis*. Oxford University Press, Oxford, UK, 157–189.

Kiers, H. A. L. 2002. Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Comput. Statist. Data Anal.* **41** 157–170.

Kiers, H. A. L., P. J. F. Groenen. 1996. A monotonically convergent algorithm for orthogonal congruence rotation. *Psychometrika* **61** 375–389.

Langbein, F. C., B. Mills, A. D. Marshall, R. R. Martin. 2001. Approximate geometric shapes. *Internat. J. Shape Model.* **7** 129–162.

Lange, K., D. R. Hunter, I. Yang. 2000. Optimization transfer using surrogate objective functions. *J. Comput. Graphics Statist.* **9** 1–20.

Rorres, C., D. G. Romano. 1997. Finding the center of a circular starting line in an ancient Greek stadium. *SIAM Rev.* **39** 745–754.

Van-Ban, L., D. T. Lee. 1991. Out-of-roundness problem revisited. *IEEE Trans. Pattern Anal.* **13** 217–223.

Van Deun, K., P. J. F. Groenen, W. J. Heiser, F. M. T. A. Busing, L. Delbeke. 2005. Interpreting degenerate solutions in unfolding by use of the vector model and the compensatory distance model. *Psychometrika* **70** 45–69.

Yeralan, S., J. A. Ventura. 1988. Computerized roundness inspection. *Internat. J. Production Res.* **26** 1921–1935.