PSYCHOMETRIKA—VOL. 65, NO. 4, 511–524
DECEMBER 2000

OPTIMAL SCALING BY ALTERNATING LENGTH-CONSTRAINED NONNEGATIVE LEAST SQUARES, WITH APPLICATION TO DISTANCE-BASED ANALYSIS

PATRICK J.F. GROENEN, BART-JAN VAN OS, AND JACQUELINE J. MEULMAN

DEPARTMENT OF EDUCATION, DATA THEORY GROUP
LEIDEN UNIVERSITY

An important feature of distance-based principal components analysis, is that the variables can be optimally transformed. For monotone spline transformation, a nonnegative least-squares problem with a length constraint has to be solved in each iteration. As an alternative algorithm to Lawson and Hanson (1974), we propose the Alternating Length-Constrained Non-Negative Least-Squares (ALC-NNLS) algorithm, which minimizes the nonnegative least-squares loss function over the parameters under a length constraint, by alternatingly minimizing over one parameter while keeping the others fixed. Several properties of the new algorithm are discussed. A Monte Carlo study is presented which shows that for most cases in distance-based principal components analysis, ALC-NNLS performs as good as the method of Lawson and Hanson or sometimes even better in terms of the quality of the solution.

Key words: nonnegative least squares, length-constraints, constrained optimization, alternating least squares, distance-based (principal components) analysis, optimal scaling, multidimensional scaling.

1. Introduction

The problem of nonnegative least squares (NNLS) arises in several applications. It is defined as the minimization of a quadratic function over the space of nonnegative elements. In this paper, we study the *length-constrained* NNLS problem (LC-NNLS), which amounts to the maximization of a linear function over the space of nonnegative elements under a length constraint. One class of problems that need to solve the LC-NNLS problem is the length-constrained monotone spline regression problem, of which monotone or isotonic regression can be viewed as a special case (Ramsay, 1988). Such problems with explicit length constraint come up in the area of optimal scaling in multidimensional scaling (see, e.g., de Leeuw & Heiser, 1977), smoothed monotone regression (Heiser, 1985), multivariate analysis with optimal scaling (see, e.g., Breiman & Friedman, 1985; Gifi, 1990; Young, de Leeuw, & Takane, 1976), and distance-based multivariate analysis with optimal scaling (Meulman, 1986, 1992). Note that for some NNLS problems fast algorithms are available that are globally optimal, such as the up-and-down-blocks algorithm of (Kruskal, 1964b) for monotone regression. In these examples, the length-constrained NNLS problem appears as a subproblem in a larger iterative process that solves the overall minimization problem. For now, we will concentrate on a distance-based principal components analysis (DB-PCA) as our overall minimization problem, because, as we will show in the next section, this problem involves a complicated exterior algorithm.

In this paper, we discuss two strategies to LC-NNLS. The first strategy consists of computing a solution to the NNLS problem without length constraint by the method of Lawson and Hanson (1974, p. 161) followed by proper normalization to impose the length constraint. This strategy always gives a globally optimal solution (de Leeuw, 1977; Gifi, 1990; Kruskal & Carroll, 1969), except in one case which is detailed in the section "Two Cases of LC-NNLS." The second strategy uses a new iterative method for LC-NNLS that never increases loss even when stopped before convergence, never violates the constraints, and at convergence yields a globally optimal

Supported by The Netherlands Organization for Scientific Research (NWO) by grant nr. 030-56403 for the "PIONEER" project "Subject Oriented Multivariate Analysis" to the third author. We would like to thank the anonymous referees for their valuable remarks that have improved the quality of this paper.

Requests for reprints should be sent to Patrick J.F. Groenen, Department of Education, Data Theory Group, Leiden University, P.O. Box 9555, 2300 RB Leiden, THE NETHERLANDS. E-Mail: groenen@fsw.leidenuniv.nl

0033-3123/2000-4/1998-0636-A $00.75/0

solution. We call this algorithm Alternating Length-Constrained Non-Negative Least Squares (ALC-NNLS). One of the reasons to come up with this alternative strategy is that it is relatively simple to program and that it is consistent with the alternating least-squares algorithms.

If the LC-NNLS problem appears as an inner optimization problem in a larger iterative scheme (as we will show in a moment), then it makes sense to use the estimates from a previous iteration as initial estimates for solving the LC-NNLS of the current iteration so that the solution may be obtained with little effort. For the method of Lawson and Hanson (1974) this has been proposed by Bro and De Jong (1997). Being an iterative algorithm, the ALC-NNLS algorithm can also make use of such good initial estimates. At convergence of the exterior algorithm, the LC-NNLS problem will be solved as well.

The purpose of this paper is to investigate whether obtaining an approximate solution of the LC-NNLS problem by ALC-NNLS is beneficial for the iterative process and for the final DB-PCA solution; in particular, whether the approximate solution of the LC-NNLS problems in the initial stages of the exterior algorithm for DB-PCA directs the exterior algorithm to an equally good or, perhaps, even better quality solution. Even though the Lawson and Hanson method solves the LC-NNLS in a finite number of steps, it is not a priori evident to us that it is the best option when it is used in a larger iterative algorithm. Note that for a related but different length-constrained optimization problem, a partial update with one inner iteration is necessary to prove convergence (Meyer, 1997; Watson, 1985).

This paper is organized as follows. First, we introduce the DB-PCA minimization problem and show how LC-NNLS is a subproblem. Then we consider properties of the LC-NNLS problem and present the ALC-NNLS algorithm for obtaining a partial update. We prove that our algorithm eventually solves the LC-NNLS problem and that it never increases the loss as defined by this problem. Then we present an illustrative example of a LC-NNLS problem arising in nonmetric multidimensional scaling showing that it can be beneficial to use the ALC-NNLS algorithm for partial updates. Next, the performance of our algorithm is investigated in a Monte Carlo study on a distance-based principal components analysis with spline transformations of the variables. We end with a discussion and some conclusions.

2. Distance-Based Principal Components Analysis

The emphasis in DB-PCA is on the optimal graphical representation of the relationships between the objects, rather than between the variables (see Gower & Hand, 1996; Krzanowski & Marriott, 1994; Meulman, 1986, 1992). In DB-PCA, the m variables are represented by the columns in the $n \times m$ matrix $\mathbf{Q}$. The aim of DB-PCA is to reconstruct the distances between these objects in the high-dimensional space $\mathbf{Q}$ as closely as possible in a low dimensional space $\mathbf{X}$. At the same time, we allow each variable k to be optimally transformed by a monotone function, that is, $\mathbf{q}_k = t_k(\mathbf{z}_k)$, where $\mathbf{z}_k$ is column k of the $n \times m$ data matrix $\mathbf{Z}$. This aim can be formalized as the minimization of the Stress loss function $\sigma^2(\mathbf{Q}, \mathbf{X})$ over $\mathbf{Q}$ and $\mathbf{X}$,

$$\sigma^2(\mathbf{Q}, \mathbf{X}) = ||D(\mathbf{Q}) - D(\mathbf{X})||^2, \tag{1}$$

where $D(\mathbf{Q})$ is the matrix containing Euclidean distances between the rows of $\mathbf{Q}$, and $||\mathbf{X}||^2$ denotes the sum of squared elements of $\mathbf{X}$, that is, tr $\mathbf{X}'\mathbf{X}$. Without loss of generality, $\mathbf{Q}$ and $\mathbf{X}$ are assumed to be column centered. As in classical principal component analysis, we let each variable have the same importance in the analysis by imposing the length constraint $\mathbf{q}_k'\mathbf{q}_k = n$ for each variable k.

We choose the $\mathbf{q}_k$'s to be monotone spline transformations of the variables, $t_k(\cdot)$ being a nondecreasing smooth piecewise polynomial of a prespecified degree. In this case, each transformed variable $\mathbf{q}_k$ can be expressed as the linear sum $\mathbf{q}_k = \mathbf{S}_k\mathbf{b}_k$, where $\mathbf{b}_k$ is a vector of nonnegative weights to be estimated and the $n \times p_k$ matrix $\mathbf{S}_k$ is an integrated spline (I-spline) basis depending only on the original variable $\mathbf{z}_k$, the degree of the spline, and a given knot sequence that defines the interval of the pieces. For more details on monotone splines we refer to

De Boor (1978) and Ramsay (1988). Note that in principal components analysis, Ramsay uses a range restriction instead of a length-constraint on the $\mathbf{q}_k$'s, thereby losing the property of equal sum of squares of the $\mathbf{q}_k$'s as in classical principal component analysis.

We prefer monotone spline transformations instead of the usual least-squares monotone transformations as emphasized in Gifi (1990), because such monotone transformations seem less suitable for variables measured on a continuous scale, and often yield a nonsmooth step function, whereas monotone splines result in a smooth transformation and control the number of parameters to be estimated. Moreover, monotone splines are easily embedded in a least-squares framework, and the least-squares monotone transformation can be considered a special case of a monotone spline transformation.

Meulman (1992) proposed a convergent algorithm in which $\mathbf{X}$ and $\mathbf{Q}$ are updated alternatingly. For ease of notation, we drop in the sequel the subscript k when referring to a variable $\mathbf{q}$. For each variable k, an unconstrained update $\overline{\mathbf{q}}$ is computed, and next the nonnegative least-squares problem with respect to $\mathbf{b}$ has to be solved, i.e.,

$$\begin{aligned} f(\mathbf{b}) &= ||\overline{\mathbf{q}} - \mathbf{S}\mathbf{b}||^2 \\ &= \overline{\mathbf{q}}'\overline{\mathbf{q}} + \mathbf{b}'\mathbf{G}\mathbf{b} - 2\mathbf{b}'\mathbf{h} \quad \text{subject to} \quad \mathbf{b} \geq \mathbf{0} \quad \text{and} \quad \|\mathbf{b}\|_{\mathbf{G}}^2 = n, \end{aligned} \tag{2}$$

where

$$\begin{aligned} \mathbf{G} &= \mathbf{S}'\mathbf{S}, \\ \mathbf{h} &= \mathbf{S}'\overline{\mathbf{q}}, \quad \text{and} \\ \|\mathbf{b}\|_{\mathbf{G}}^2 &= \mathbf{b}'\mathbf{G}\mathbf{b}. \end{aligned}$$

We assume that $\mathbf{S}$ is of full rank, which is generally true for I-spline bases. Thus, in each iteration, m LC-NNLS problems have to be solved. To retain overall monotone convergence, it is not necessary to solve (2) completely, but to find an update of $\mathbf{b}$ for which (2) is not larger than the previous iteration.

3. Length-Constrained NNLS (LC-NNLS)

LC-NNLS fixes the length of $\mathbf{b}$ to $\|\mathbf{b}\|_{\mathbf{G}}^2 = n$, so that (2) becomes

$$f(\mathbf{b}) = c + n - 2\mathbf{b}'\mathbf{h} \quad \text{subject to} \quad \mathbf{b} \geq \mathbf{0} \quad \text{and} \quad \|\mathbf{b}\|_{\mathbf{G}}^2 = n, \tag{3}$$

where $c = \|\overline{\mathbf{q}}\|^2$. Below we analyze (3) using convex analysis, and show that two cases of LC-NNLS can be distinguished; we also show under which conditions (3) has a unique solution.

3.1. *Uniqueness at the Minimum of LC-NNLS*

Here we prove that (3) has a unique solution under a mild condition. Consider the sets

$$X = \{\mathbf{b} \in \mathbb{R}^p \mid \|\mathbf{b}\|_{\mathbf{G}}^2 \leq n\}, \tag{4}$$

$$\widehat{X} = \{\mathbf{b} \in \mathbb{R}^p \mid \|\mathbf{b}\|_{\mathbf{G}}^2 = n\}, \tag{5}$$

$$Y = \{\mathbf{b} \in \mathbb{R}^p \mid \mathbf{b}'\mathbf{w} \geq n\}, \quad \text{and} \tag{6}$$

$$P = \{\mathbf{b} \in \mathbb{R}^p \mid \mathbf{b} \geq \mathbf{0}\}, \tag{7}$$

where $\mathbf{w}$ has elements $w_i = (ng_{ii})^{1/2}$ for $i = 1, \ldots, p$. Set $\widehat{X}$ defines all $\mathbf{b}$'s that satisfy the length constraint and set X defines all $\mathbf{b}$'s with equal or smaller length than the length constraint. Thus set $\widehat{X}$ defines the boundary of X. Set Y is one of the two halfspaces separated by the subspace $\mathbf{b}'\mathbf{w} = n$ that cuts through those points where the ellipsoid of $\widehat{X}$ cuts the axes. The last

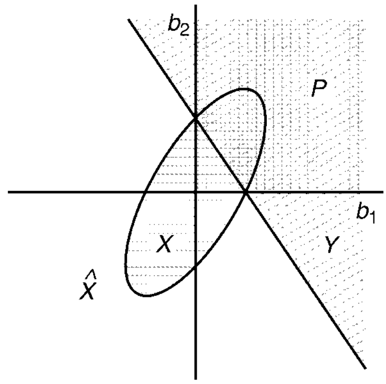

FIGURE 1.
An example in $\mathbb{R}^2$ of the sets X, $\widehat{X}$, P, and Y in LC-NNLS. Note that $\widehat{X}$ refers to the boundary of the ellipse.

set P defines all points $\mathbf{b}$ in the positive orthant. An example in $\mathbb{R}^2$ of the sets X, $\widehat{X}$, P, and Y is given in Figure 1.

Using these sets, LC-NNLS can be expressed as minimizing $f(\mathbf{b})$ over $\mathbf{b} \in \widehat{X} \cap P$. We now explain that the minimum of $-\mathbf{b}'\mathbf{h}$ over $\mathbf{b}$ in the convex set $Z = X \cap Y \cap P$ is equal to the minimum of $f(\mathbf{b})$ in (3). Let the extremal points of a convex set C be those points in C that cannot be expressed as a convex combination of two other points in C (Rockafellar, 1970, p. 162). For example, P only has the origin as extremal point, but all the points in $\widehat{X}$ are extremal points of X. Since the sets X, Y, and P are convex, the intersection set $Z = X \cap Y \cap P$ is convex, too. The extremal points of Z coincide with the points in $\widehat{X} \cap P$, which is the set satisfying the constraints in (3). Theorem 32.3 in Rockafellar (p. 344) implies that the minimum of a linear function over a bounded closed convex set is obtained at an extremal point. This explains that the minimum of $-\mathbf{b}'\mathbf{h}$ over $\mathbf{b}$ in the convex set Z is equal to the minimum of LC-NNLS.

The minimum point is unique if two conditions are met. First, we require positive definiteness of $\mathbf{G}$, so that X is not in a subspace of $\mathbb{R}^p$. Second, we require that $\mathbf{h} \neq -c\mathbf{w}$ (for some $c > 0$). If $\mathbf{h} = -c\mathbf{w}$, then any of the p extremal points of the polyhedral set $P \cap \{\mathbf{b}'\mathbf{w} = n\}$ yields the same $f(\mathbf{b})$ while satisfying the length constraint. If these two conditions are met, then the solution for LC-NNLS is obtained at an extremal point, which is by definition unique.

3.2. Two Cases of LC-NNLS

We distinguish two cases in the minimization of (3):

(i) all h_i's are negative;
(ii) there exists at least one positive h_i.

For Case (i) a direct solution exists, but for Case (ii) a special algorithm to minimize (3) is needed, which will be discussed in the next section. We now continue with Case (i) and show that the function

$$f_2(\mathbf{b}) = c + n - 2\mathbf{b}'\mathbf{h} \quad \text{with} \quad \mathbf{b} \in Y \cap P \tag{8}$$

reaches its minimum at one of the vertices of the polyhedron defined by $Y \cap P$, that is, for $b_i = (n/g_{ii})^{1/2}$ and all other $b_{j \neq i} = 0$. The vertex that should be nonzero, is the one for which $f_2(\mathbf{b})$ is minimal, that is,

$$\text{argmin}_i - h_i(n/g_{ii})^{1/2}. \tag{9}$$

Since $(Y \cap P) \supset Z$, this solution also minimizes $f(\mathbf{b})$ in (3). Note that $Y \cap P$ defines a polyhedral set. Rockafellar (1970, Corollary 32.3.4, p. 345) states that if $f(\mathbf{b})$ is bounded below on a polyhedral set, then the minimum is obtained at an extremal point. For negative $\mathbf{h}$, boundedness from below is implied, because $-\mathbf{h}'\mathbf{b}$ is positive for all feasible $\mathbf{b}$. Checking the minimum values of $f(\mathbf{b})$ at the extremal points as in (9) gives the minimum.

4. Algorithms for LC-NNLS

In this section, we discuss two computational methods that provide a solution for Case (ii) of LC-NNLS. The first method is to compute an unconstrained minimum (i.e., without length constraint) and then apply the normalization constraint such that $\|\mathbf{b}\|^2_{\mathbf{G}} = n$ (de Leeuw, 1977; Gifi, 1990; Kruskal & Carroll, 1969). For NNLS this minimum can be found by the method of Lawson and Hanson (1974), which gives an exact solution in a finite number of steps. In the sequel, we shall refer to this method as LH-NNLS. The second method is an improvement of the approach used in Meulman (1992). Her approach consists of cyclically updating one $b_i \geq 0$ at a time, disregarding the length constraint, evaluating (2) or (3) after $\mathbf{b}$ is updated for all variables, and continuing with a subsequent round of updates until the value of (2) or (3) does not increase. At that point the length constraint can be applied. Here, we improve the approach of Meulman by solving for the nonnegativity constraints and the length constraint simultaneously. We call this new algorithm Alternating Length-Constrained Non-Negative Least Squares (ALC-NNLS) and it is described below.

4.1. ALC-NNLS

In this section, we first propose the ALC-NNLS algorithm. To show why it works, we reformulate the LC-NNLS problem, and then prove that $f(\mathbf{b})$ in every step of ALC-NNLS is reduced or remains the same. The estimate of $\mathbf{b}$ at iteration ℓ is denoted by $\mathbf{b}^{(\ell)}$; $\|\mathbf{b}\|_{\mathbf{G}}$ denotes $(\mathbf{b}'\mathbf{G}\mathbf{b})^{1/2}$.

The ALC-NNLS algorithm is summarized by the following steps, where ϵ is a preset small value and $\ell_{\max}$ the preset maximum number of iterations.

1. Choose some $\bar{\mathbf{b}} \neq \mathbf{0}$. If $\mathbf{h}'\bar{\mathbf{b}} < 0$ then $\bar{b}_i := \max(0, h_i)$ for all i.
2. If $\bar{\mathbf{b}} = \mathbf{0}$ then compute the Case (i) solution by (9) and stop.
3. Set $\mathbf{b}^{(0)} := n^{1/2}\bar{\mathbf{b}}/\|\bar{\mathbf{b}}\|_{\mathbf{G}}$. Set $\ell := 0$.
4. $\quad \ell := \ell + 1$.
5. $\quad$ Set $a := \mathbf{h}'\mathbf{b}^{(\ell-1)}/n$, and $\bar{\mathbf{b}} := \mathbf{b}^{(\ell-1)}$.
6. $\quad$ For each i, set $\bar{b}_i := \max\left(0, (ag_{ii})^{-1}\left[h_i - a\sum_{j\neq i} g_{ij}\bar{b}_j\right]\right)$.
7. $\quad$ Set $\mathbf{b}^{(\ell)} := n^{1/2}\bar{\mathbf{b}}/\|\bar{\mathbf{b}}\|_{\mathbf{G}}$.
8. If $f(\mathbf{b}^{(\ell-1)}) - f(\mathbf{b}^{(\ell)}) > \epsilon$ or $\ell < \ell_{\max}$ go to 4.
9. Stop.

In the remainder of this section, we explain how this algorithm arises and prove that $f(\mathbf{b})$ never increases. We start by showing that LC-NNLS is equivalent to the maximization of the function

$$g(\mathbf{b}) = \frac{(\mathbf{b}'\mathbf{h})^2}{\|\mathbf{b}\|^2_{\mathbf{G}}} \quad \text{subject to} \quad \mathbf{b} \geq \mathbf{0} \quad \text{and} \quad \mathbf{b}'\mathbf{h} \geq 0, \tag{10}$$

which does not have a length constraint. We will show below that our algorithm always satisfies the restriction $\mathbf{b}'\mathbf{h} > 0$.

The length constraint is automatically satisfied by substituting $\mathbf{b} = \bar{\mathbf{b}}(n^{1/2}/\|\bar{\mathbf{b}}\|_{\mathbf{G}})$ for all $\bar{\mathbf{b}} \neq \mathbf{0}$, so that $\mathbf{b}'\mathbf{G}\mathbf{b} = \bar{\mathbf{b}}'\mathbf{G}\bar{\mathbf{b}}(n/\bar{\mathbf{b}}'\mathbf{G}\bar{\mathbf{b}}) = n$. Inserting this choice of $\mathbf{b}$ in $f(\mathbf{b})$ yields

$$
\begin{aligned}
f(\mathbf{b}) = f\left(\frac{n^{1/2}\bar{\mathbf{b}}}{\|\bar{\mathbf{b}}\|_{\mathbf{G}}}\right) &= n - 2n^{1/2}\frac{\bar{\mathbf{b}}'\mathbf{h}}{\|\bar{\mathbf{b}}\|_{\mathbf{G}}} + c \\
&= n - 2n^{1/2}g^{1/2}(\bar{\mathbf{b}}) + c \quad \text{subject to} \quad \bar{\mathbf{b}} \geq \mathbf{0}.
\end{aligned} \tag{11}
$$

Apart from some constants, the difference between (10) and (11) is that the latter uses the square root of $g(\mathbf{b})$ and the former does not. However, such monotone transformations do not change the extremal points of the function, provided $\mathbf{b}'\mathbf{h} \geq 0$. Under the condition $\mathbf{b}'\mathbf{h} \geq 0$, LC-NNLS is equivalent to the maximization of (10).

The next step is to find an update that decreases $-g^{1/2}(\mathbf{b})$. Define the auxiliary function

$$
f_2(\mathbf{b}) = \tfrac{1}{2}a\mathbf{b}'\mathbf{G}\mathbf{b} - \mathbf{h}'\mathbf{b} + \tfrac{1}{2}\mathbf{h}'\mathbf{b}^{(\ell-1)} \tag{12}
$$

with $a = \mathbf{h}'\mathbf{b}^{(\ell-1)}/\|\mathbf{b}^{(\ell-1)}\|^2_{\mathbf{G}}$. Now we prove that any update $\mathbf{b}^{(\ell)}$ for which $f_2(\mathbf{b}^{(\ell)}) \leq 0$ ensures that $-g^{1/2}(\mathbf{b}^{(\ell)}) \leq -g^{1/2}(\mathbf{b}^{(\ell-1)})$. In this proof, we make use of Dinkelbach's (1967) approach to minimize a ratio of functions. We assume that $\|\mathbf{b}\|_{\mathbf{G}} > 0$, and that $\mathbf{b}'\mathbf{h} > 0$.

Some algebraic manipulation with the inequality $(\|\mathbf{b}\|_{\mathbf{G}} - \|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}})^2 \geq 0$ gives

$$
\|\mathbf{b}\|_{\mathbf{G}} \leq \frac{1}{2}\frac{\|\mathbf{b}\|^2_{\mathbf{G}}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}} + \frac{1}{2}\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}} \tag{13}
$$

which becomes an equality if $\mathbf{b} = \mathbf{b}^{(\ell-1)}$. Multiplying both sides of (13) by the positive value $\mathbf{h}'\mathbf{b}^{(\ell-1)}/\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}$ gives

$$
\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}}\|\mathbf{b}\|_{\mathbf{G}} \leq \frac{1}{2}\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}}\|\mathbf{b}\|^2_{\mathbf{G}} + \frac{1}{2}\mathbf{h}'\mathbf{b}^{(\ell-1)}. \tag{14}
$$

Combining (12) and (14) gives

$$
\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}}\|\mathbf{b}\|_{\mathbf{G}} - \mathbf{h}'\mathbf{b} \leq \frac{1}{2}\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}}\|\mathbf{b}\|^2_{\mathbf{G}} + \frac{1}{2}\mathbf{h}'\mathbf{b}^{(\ell-1)} - \mathbf{h}'\mathbf{b} = f_2(\mathbf{b}). \tag{15}
$$

Note that $f_2(\mathbf{b}^{(\ell-1)}) = 0$ and that (15) becomes an equality if $\mathbf{b} = \mathbf{b}^{(\ell-1)}$. If we can find an update $\mathbf{b}^{(\ell)}$ for which $f_2(\mathbf{b}^{(\ell)}) \leq 0$, then we must have

$$
\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}}\|\mathbf{b}^{(\ell)}\|_{\mathbf{G}} - \mathbf{h}'\mathbf{b}^{(\ell)} \leq 0. \tag{16}
$$

The next step in the proof forms the heart of Dinkelbach's approach. Dividing (16) by the positive value $\|\mathbf{b}^{(\ell)}\|_{\mathbf{G}}$ gives

$$
\begin{aligned}
\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}} - \frac{\mathbf{h}'\mathbf{b}^{(\ell)}}{\|\mathbf{b}^{(\ell)}\|_{\mathbf{G}}} &\leq 0, \\
-\frac{\mathbf{h}'\mathbf{b}^{(\ell)}}{\|\mathbf{b}^{(\ell)}\|_{\mathbf{G}}} &\leq -\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}}, \\
-g^{1/2}(\mathbf{b}^{(\ell)}) &\leq -g^{1/2}(\mathbf{b}^{(\ell-1)}),
\end{aligned}
$$

which proves that an update $\mathbf{b}^{(\ell)}$ with $f_2(\mathbf{b}^{(\ell)}) \leq 0$ guarantees $-g^{1/2}(\mathbf{b}^{(\ell)}) \leq -g^{1/2}(\mathbf{b}^{(\ell-1)})$.

It remains to be proven that in our algorithm $\mathbf{h}'\mathbf{b}^{(\ell)} > 0$, so that maximizing $g(\mathbf{b})$ and $g^{1/2}(\mathbf{b})$ are equivalent. Suppose that $\mathbf{b}^{(0)}$ is such that $\mathbf{h}'\mathbf{b}^{(0)} > 0$. (Our algorithm satisfies this condition in Step 1.) Also suppose that $-g^{1/2}(\mathbf{b}^{(\ell)}) \leq -g^{1/2}(\mathbf{b}^{(\ell-1)})$ is satisfied in every iteration, then the sequence

$$-\frac{\mathbf{h}'\mathbf{b}^{(\ell)}}{\|\mathbf{b}^{(\ell)}\|_{\mathbf{G}}} \leq -\frac{\mathbf{h}'\mathbf{b}^{(\ell-1)}}{\|\mathbf{b}^{(\ell-1)}\|_{\mathbf{G}}} < 0,$$

implies that $\mathbf{h}'\mathbf{b}^{(\ell)} > 0$, since $\|\mathbf{b}^{(\ell)}\|_{\mathbf{G}} > 0$ by assumption. Therefore, the condition $\mathbf{h}'\mathbf{b}^{(\ell)} > 0$ is satisfied in our algorithm in every iteration ℓ. (A different proof that minimizing $f_2(\mathbf{b})$ in each iteration reduces, or never increases, $-g(\mathbf{b})$ can be given by using the iterative majorization result of Kiers, 1995.)

Thus, for obtaining a nonincreasing series of function values $f(\mathbf{b})$ of the LC-NNLS problem it is sufficient that in each iteration an update is found for which the quadratic function $f_2(\mathbf{b})$ is decreased (or remains equal). This step should be followed by a normalization step (Steps 3 and 7) in the ALC-NNLS algorithm. Note that in the minimization of $f_2(\mathbf{b})$ no length constraint is active so that $f_2(\mathbf{b})$ defines a nonnegative least squares problem.

The core of the ALC-NNLS algorithm, Step 6, is to reduce $f_2(\mathbf{b})$ over one b_i at a time subject to $b_i > 0$. This can be seen as follows. Let us rewrite $f_2(\mathbf{b})$ as

$$f_2(\mathbf{b}) = \tfrac{1}{2}a\sum_i b_i^2 g_{ii} + \sum_i b_i \sum_{j\neq i} g_{ij}b_j - \sum_i b_i h_i + \tfrac{1}{2}\sum_i b_i^{(\ell-1)}h_i \tag{17}$$

and express the derivative to b_i as

$$\frac{\partial f_2(\mathbf{b})}{\partial b_i} = a g_{ii} b_i - \left(h_i - a\sum_{j\neq i} g_{ij}b_j\right). \tag{18}$$

The derivative is equal to zero for $b_i = (a g_{ii})^{-1}(h_i - a\sum_{j\neq i} g_{ij}b_j)$. Now, b_i takes this value if it is nonnegative, else b_i is set to zero, which is exactly the definition of Step 6 in the algorithm. Of course, one could also repeat Step 6 several times, so that a better value of $f_2(\mathbf{b})$ in (17) is obtained. In fact, some experimentation of ALC-NNLS in DB-PCA showed that setting $\ell_{\max} = 1$ and repeating Step 6 one or more times is to be preferred over doing Step 6 only once with $\ell_{\max} \geq 1$.

5. Solving LC-NNLS by Small Steps: An Illustrative Example

In this section, we present an illustrative example that supports the idea that not completely solving the LC-NNLS problem and taking only a few steps in the right direction gives a good quality solution faster. The example we use is taken from multidimensional scaling (MDS), a technique that is used to represent proximities between pairs of objects by distances between points in a low dimensional space representing those objects (see, e.g., Borg & Groenen, 1997; Kruskal, 1964a). Instead of approximating the proximities directly, it is custom in MDS to allow the proximities to be transformed optimally, much in the same way as variables are transformed optimally in DB-PCA. The loss function to be minimized is

$$\sigma^2_{\mathrm{MDS}}(\widehat{\mathbf{D}}, \mathbf{X}) = \|\widehat{\mathbf{D}} - D(\mathbf{X})\|^2, \tag{19}$$

where $\widehat{\mathbf{D}}$ is the symmetric matrix with zero diagonal of optimally transformed proximities also called pseudo-distances. Note that (19) is closely related to (1), where the distances of $\mathbf{X}$ approximate different type of data . To avoid the trivial solution $\widehat{\mathbf{D}} = \mathbf{0}$ and $\mathbf{X} = \mathbf{0}$, the length constraint $\|\widehat{\mathbf{D}}\| = c$ with $c > 0$ is imposed (de Leeuw & Heiser, 1977). In this example, the pseudo-distances are restricted to be monotone spline transformations of the proximities. Thus, the minimization of $\sigma^2_{\mathrm{MDS}}(\widehat{\mathbf{D}}, \mathbf{X})$ has LC-NNLS as a subproblem in a larger iterative scheme.

For our example, we constructed proximities by taking the Euclidean distances between the rows of an error perturbed data matrix (see the Appendix) of $n = 25$ objects in two dimensions

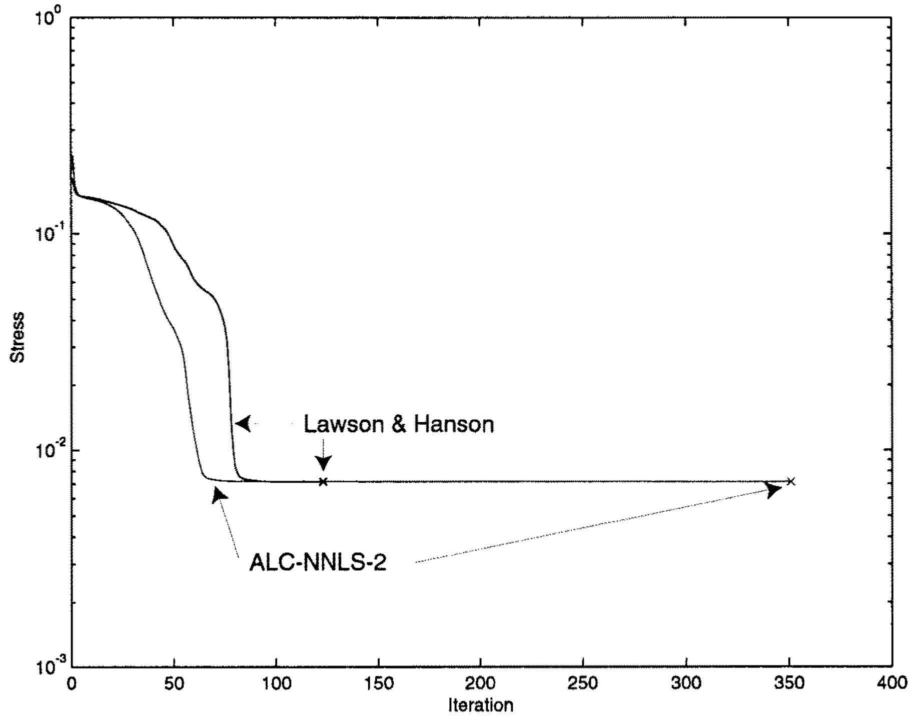

FIGURE 2.
Stress values of the MDS example as a function of the iteration for three methods for solving LC-NNLS: LH-NNLS and ALC-NNLS-2 with two inner iterations.

rotated and expanded to 10 dimensions and finally perturbed by 25% error. This leads to $n(n-1)/2 = 300$ different proximities. The monotone spline transformation was set to have three interior knots and be of degree two. The dimensionality of $\mathbf{X}$ was set to two. The minimization process was stopped if the drop in two subsequent Stress values was less than 10^{-8}.

To investigate whether it is fruitful to solve the LC-NNLS problem partially (by using ALC-NNLS instead of LH-NNLS) consider the Stress value (19) as a function of the iteration for two different methods for solving LC-NNLS: LH-NNLS and ALC-NNLS-2 with two inner iterations. Figure 2 shows the results for this example. During the first part of the MDS iterations (say up to iteration 90), the Stress for ALC-NNLS-2 drops faster than that of LH-NNLS. In the last part of the iterations, ALC-NNLS-2 needs more iterations to reach the convergence criterion than LH-NNLS.

We investigated how fast the methods approximate the $\widehat{\mathbf{D}}$ at their final iteration. It turned out that during the first part of the iterations the steps towards the final $\widehat{\mathbf{D}}$ are faster for ALC-NNLS-2 than for LH-NNLS.

From this illustrative example two things can be concluded. First, the use of ALC-NNLS can make larger steps towards the final solution during the first part of the iterations than the use of LH-NNLS. Second, in the last part of the iterative process, ALC-NNLS with only a few inner iterations may reach convergence more slowly than LH-NNLS if the convergence criterion is rather strict.

6. A Monte Carlo Study

A Monte Carlo study was conducted to investigate the performance of the ALC-NNLS strategy relative to LH-NNLS for solving the LC-NNLS problem in a distance-based principal components analysis (DB-PCA). The LC-NNLS problem appears in the transformations of the variables by the choice of second-degree I-splines. The number of inner iterations in the ALC-NNLS strategy was set to one, two, three, five, ten, or dependent on convergence with a maximum of 50 iterations. The spline weights for the ALC-NNLS algorithm were initialized by a positive constant, such that the length constraint was satisfied. We used the Fortran subroutines for LH-NNLS that are available in Lawson and Hanson (1974). The exterior iterative scheme that minimizes Stress was stopped whenever the drop in *normalized* Stress, that is, $(n^2m)^{-1}\sigma^2(\mathbf{Q}, \mathbf{X})$, between two subsequent estimates was less than 10^{-5}. This choice seems reasonable because the coordinates will be determined within two to three digits accuracy and it imitates the stopping criterion for a typical analysis. This stopping criterion (as many other criteria) cannot exclude the possibility that the algorithm stops prematurely in some cases.

6.1. Method

6.1.1. Dependent Variables

The performance of the strategies was studied by comparing DB-PCA solutions for constructed multivariate datasets on two types of criteria. First, we were interested whether there is a systematic difference in the quality of the solution obtained by different LC-NNLS strategies. At convergence, the DB-PCA algorithm does not guarantee a global minimum but at most a local minimum. Therefore, we are interested whether the likelihood of attaining a (candidate) global minimum differs among strategies. This criterion was operationalized by comparing the average Stress among the strategies. Second, we are interested in the convergence behavior of the strategies. The efficiency of convergence was measured by observing the number of exterior iterations needed to converge. This measure is considered to be reasonable, because the workload of solving the LC-NNLS problems is generally smaller than the other computations required in a DB-PCA iteration.

6.1.2. Design

The dependent variables were used in a mixed within-subjects between-subjects design, taking the datasets to be analyzed as "subjects." The seven strategies (LH-NNLS, ALC-NNLS-1, −2, −3, −5, −10, −50) constituted the first within-subject factor.

We varied several design factors that might influence the importance of the LC-NNLS problem in minimizing Stress. These factors are: (a) number of objects, $n = 50, 100$; (b) number of variables, $m = 5, 10, 20$; (c) number of interior knots in the spline transformations, $k = 3, 9$; and (d) error added to the data, $\varepsilon = 10\%, 25\%, 40\%$. The number of objects controls for the size of the problem of estimating $\mathbf{X}$ and $\mathbf{Q}$, which is of different order of magnitude compared to the LC-NNLS problems. The number of variables determines the number of LC-NNLS problems to be solved at each major iteration. The number of knots used in the spline transformations influences the size of the LC-NNLS problem to be solved. To ensure that transformations are required, we added random noise to the variables in each dataset. In the Appendix we describe how the error perturbed datasets were constructed. The four non-strategy factors lead to an overall design, where seven strategies are tested in 36 cells. The number of knots in the spline transformation constitutes the second within-subject factor. Because the variability in Stress values due to the design factors is not known a priori, a statistical decision on the number of replications needed in each cell can not easily be made in advance. In a trial set of 50 replications, the main-effects of the within-subjects factors proved to be relatively small. We decided to have 150 replications in each cell to be on the safe side. In total, 2700 datasets ($2 \times 3 \times 3 \times 150$) were analyzed using seven different strategies under the two transformation conditions (number of interior knots). The pseudo-random numbers were generated using the linear congruential generator of the MATH77 package, which is known to have good characteristics (see Jones & Seaton, 1994, sec. 3.1-5). In each cell, the 150 replications were generated in three sets, using three seeds, one for each set of 50 replications.

6.2. *Results*

All solutions of the DB-PCA analyses conducted in the design showed a good recovery of the "true" two-dimensional structure **H**. The average Tucker coefficients of congruence between the distances of the recovered configuration **X** and the constructed ideal configuration **H** were .99, .91, .85 for error-levels 10%, 25%, 40%, respectively. The Tucker congruence coefficient reflects the degree to which two variables are identical up to a positive multiplicative transformation.

The average Stress of the solution—our measure of quality—proved to be dependent on the NNLS-strategy used. A repeated-measures ANOVA showed the main and all but one first-order interaction effect for the strategy factor to be significant ($P < 0.001$) in average univariate tests and in multivariate tests. The high levels of significance may to a large extent be explained by the high number of replications. The effects are stable but relatively small and thus need to be interpreted with some care. Small effects are not surprising, since the LC-NNLS is only a part of the total optimization problem and the strategies search for the same optimum of the Stress-function for a given dataset. The first-order interaction effects were much smaller than the main effects of the strategy factor. On average, the ALC-NNLS strategy with two inner iterations found the smallest Stress-value (see Table 1). This strategy was particularly better in combination with the 9-knot condition and in combination with the 20-variables condition, but there was no interaction with the number of objects. The interaction effect between the strategy factor and the error-level factor showed a preference for LH-NNLS in the 40% error-level condition. Apparently, the LH-NNLS strategy is more suitable in high error conditions, where the spline-transformation constitutes a larger part of the overall decrease in Stress.

To investigate the convergence efficiency conditional on the quality of the solution, the same ANOVA model as above was used with the number of iterations as dependent variable correcting for the Stress value. This model did not find a significant main effect of the strategies, indicating that the average number of iterations needed by the strategies did not differ when corrected for the quality of the solution.

TABLE 1.
Average Stress values for main effects of the LC-NNLS strategy, and interactions with error level ε, number of objects n, number of variables m, and number of knots k

		ALC-NNLS					
	LH-NNLS	1	2	3	5	10	50
$\varepsilon = 0.10$	.0537	.0552	.0534	.0532	.0535	.0537	.0537
$\varepsilon = 0.25$	.0836	.0845	.0834	.0836	.0837	.0836	.0836
$\varepsilon = 0.40$	.1065	.1072	.1065	.1066	.1065	.1065	.1065
$n = 50$	.0786	.0798	.0785	.0785	.0787	.0786	.0786
$n = 100$	.0838	.0848	.0836	.0837	.0838	.0838	.0838
$m = 5$	.0660	.0676	.0659	.0659	.0660	.0659	.0660
$m = 10$	.0832	.0842	.0830	.0831	.0832	.0832	.0832
$m = 20$	.0946	.0952	.0943	.0944	.0945	.0945	.0946
$k = 3$	.0878	.0889	.0878	.0877	.0878	.0878	.0878
$k = 9$	.0747	.0757	.0744	.0746	.0747	.0747	.0747
Total	.0812	.0823	.0811	.0811	.0812	.0812	.0812

TABLE 2.
Average number of DB-PCA iterations using the LH-NNLS or the ALC-NNLS-2 strategy

		Average # iterations		
		LH-NNLS	ALC-NNLS-2	# Repl
(a)	No difference	129	130	849
(b)	LH-NNLS better	171	153	853
(c)	ALC-NNLS-2 better	133	148	1303

To exemplify the effect of the strategy-quality interaction on the average number of DB-PCA iterations, we compare in Table 2 the average number of iterations of the LH-NNLS and ALC-NNLS-2 strategies, the latter being the best of the ALC-NNLS strategies. We selected three groups of runs based on the difference between the two strategies in terms of Stress: (a) a group with differences smaller than $2 \cdot 10^{-5}$, (b) a group with differences larger than 10^{-4} in favor of LH-NNLS, and (c) a group with differences larger than 10^{-4} in favor of ALC-NNLS-2. For equally good solutions, the average number of iterations did not differ substantially between LH-NNLS and ALC-NNLS-2. For group (b), where NNLS-LH performed better, its average number of iterations is larger, and the reverse situation occurs in group (c), where ALC-NNLS-2 performed better. The conclusion is that the number of iterations is determined by the quality of the solution and the design factors, but not by the strategies.

To see if there is a systematic difference in convergence behavior of the strategies, we selected three strategies (LH-NNLS, ALC-NNLS-2, and ALC-NNLS-3) and compared the Stress against the iteration. The iteration number seems a fair measure to compare the computational load. Measuring CPU times of the LC-NNLS update in the larger problem arising in multidimensional scaling (see the previous section) revealed that the fastest possible implementations of LH-NNLS using good initial estimates (Bro & De Jong, 1997) and ALC-NNLS with several inner iterations needed about the same CPU time.

We selected the cells from the study that correspond to $k = 3, 9$, $m = 20$, $n = 50$, and $\varepsilon = 10\%, 25\%, 40\%$. For each combination, 50 gauges were constructed, yielding a total of 300 different data sets. The number of iterations in each run was fixed to 1200. For each of the three strategies it was determined in every iteration how often the strategy was close to the smallest Stress value in that iteration. By close, we mean here that the Stress should not exceed 10^{-6} plus the lowest Stress found. Figure 3 shows the proportion of lowest Stress against the iteration for the three strategies. Until iteration 50 or so, LH-NNLS obtains the lowest Stress in the majority of the cases. Approximately from iteration 100 to 900, ALC-NNLS-2 finds the lowest Stress values in about 60 to 70% of the cases. After iteration 900, LH-NNLS reaches the lowest Stress slightly more often than ALC-NNLS. We also see that from iteration 800 all three strategies perform quite well, suggesting that they all move towards the same optimum. These effects were strongest for the $k = 9$ condition. From Figure 3 it may be concluded that for the LC-NNLS problems arising in DB-PCA the ALC-NNLS-2 strategy is beneficial in the middle stage in reaching lower Stress values faster. A fast start can be obtained by LH-NNLS. Towards the end, all three strategies perform equivalent.

7. Discussion and Conclusions

We have studied the length-constrained nonnegative least squares (LC-NNLS) problem in some detail. We identified the case in which an explicit solution is available. The other case can be solved by the method of Lawson and Hanson (1974) followed by proper normalization, or by a new algorithm, called alternating length-constrained nonnegative least squares (ALC-NNLS), for solving the length-constrained problem in an iterative manner. For LC-NNLS problems arising in larger iterative schemes, we investigated whether it could be beneficial not to solve the LC-

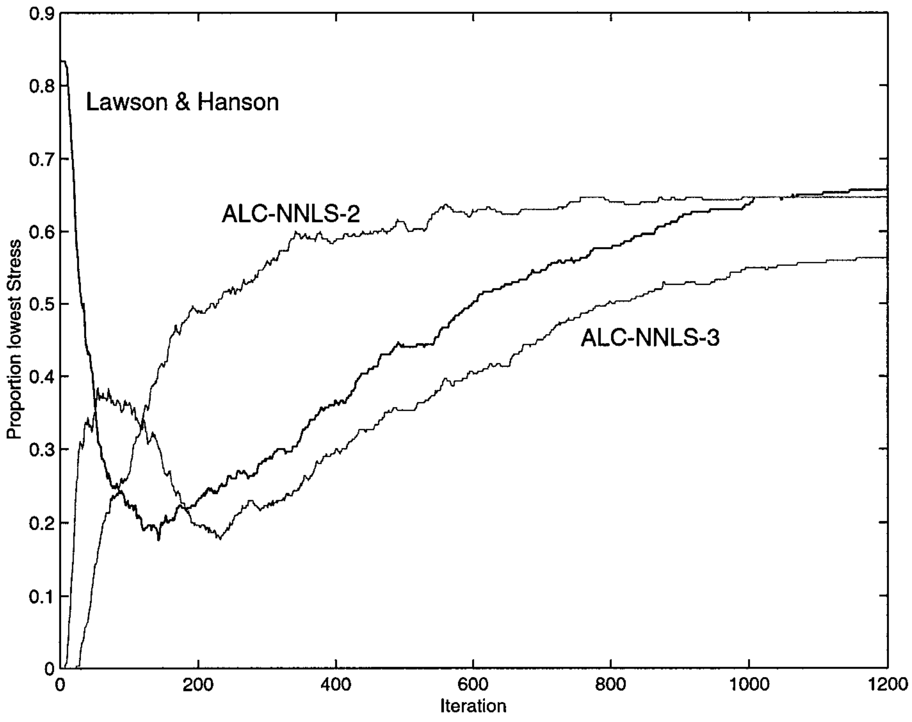

FIGURE 3.
For 300 data sets the Stress value was recorded for three strategies during the first 1200 iterations. Then for every data set and at every iteration it was determined whether the strategy was close to the lowest Stress value found at that iteration by any of the three strategies. For each strategy the proportion of data sets for which this condition holds is plotted as a function of the iteration.

NNLS problem completely, but to take only one or more steps in the right direction. However, if the LC-NNLS problem is not embedded in a larger iterative scheme but appears on its own, we recommend the strategy of Lawson and Hanson (1974) followed by proper normalization, since it yields an exact solution.

We investigated the LC-NNLS problem occurring in distance-based principal components analysis (DB-PCA) using I-spline transformations. A Monte Carlo study compared ALC-NNLS strategies to the Lawson and Hanson (1974) approach. With respect to the quality of the DB-PCA solutions, ALC-NNLS with two or three inner iterations is the most preferable alternative, and doing only one inner iteration is the least preferred alternative. This result might warrant revision of the ALS-approach advocated in Gifi (1990), which is using only one inner iteration. In the majority of the cases, ALC-NNLS with two or three inner iterations performs better than the strategy of Lawson and Hanson, except in high error conditions. Comparing the Stress values against the iteration revealed that ALC-NNLS with two or three inner iterations is close to the lowest Stress value in about 60 to 70% of the cases during the middle part of the iterative process. At the very beginning, LH-NNLS is often better, whereas towards the end, ALC-NNLS and LH-NNLS behave similarly, with LH-NNLS reaching the lowest Stress slightly more often. We conjecture that not completely solving the LC-NNLS problem in each DB-PCA iteration prevents the DB-PCA algorithm from taking inappropriate paths during the middle part of the iterative process. If a high precision or a strong convergence criterion is needed, then the ALC-NNLS strategies will probably reach convergence slower than the Lawson and Hanson strategy, since the latter guarantees an exact solution. For such cases, a hybrid method might be useful, consisting of ALC-NNLS steps during the earlier exterior iterations and switching to the Lawson and Hanson strategy or ALC-NNLS-50 at a later stage.

The algorithm proposed in this paper can also be used for solving other least-squares problems with complex restrictions under an explicit length constraint, such as the bounded monotone regression problem of Verboon (1994). It is not clear at this point whether our results generalize to other exterior algorithms. We expect that our method becomes more effective in situations where the LC-NNLS problem has a considerable number of interior knots, such as in suitable transformations of proximities in multidimensional scaling.

A stand alone program called PIONEER for doing DB-PCA including spline transformations can be found at the PIONEER home page `http://www.fsw.leidenuniv.nl/www/w3_data/pioneer/pioneer.htm`. The program has an SPSS-like command syntax. For more details about the program, see Groenen, Commandeur, and Meulman (1997) or the manual that can be found at the PIONEER home page.

Appendix: Error Perturbed Data

The error perturbed data $\mathbf{Z}$ need to fulfill the following requirements: (1) all variables should have mean zero and equal sum of squares, (2) the data matrix should be of rank two, and (3) error must be added so that transformations of the data may be necessary. $\mathbf{Z}$ was constructed as follows. We first compute a 100×2 matrix $\mathbf{H}$ with elements $h_{ij} \sim N(0, 1)$, where $N(0, 1)$ denotes the normal distribution with mean zero and variance one. Then, $\mathbf{H}$ was rotated and expanded to m dimensions using a pseudo-random rotation-expansion matrix $\mathbf{A}$ with rows on the unit circle, i.e, $\mathbf{A}'\mathbf{A} = c\mathbf{I}$, $\text{Diag}(\mathbf{AA}') = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. These restrictions ensure that the distances between the rows of $\mathbf{H}$ are the same as those of $\mathbf{HA}'$ up to a constant factor, and the unit circle constraint additionally ensures that the columns of $\mathbf{HA}'$ have approximately the same sum of squares. (We only have approximate equality of the column sum of squares of $\mathbf{HA}'$ because $\mathbf{H}$ is the result of a sampling process.) $\mathbf{A}$ was created as follows. The procedure described below is especially efficient for creating rank two data. (For higher rank data an alternative method is available.) If m was even then $m/2$ random values ϕ_i are drawn with $0 \le \phi_i < 2\pi$ and the rows of $\mathbf{A}$ are filled in pairs:

$$\begin{aligned} a_{i,1} &= \cos\phi_i, \\ a_{i,2} &= \sin\phi_i, \\ a_{m/2+i,1} &= -\sin\phi_i, \\ a_{m/2+i,2} &= \cos\phi_i. \end{aligned}$$

If m is odd, then the first $m - 2$ rows are filled as for m even. Then

$$\begin{aligned} a_{m-1,1} &= a_{(m-1)/2,1}\cos 2\pi/3 - a_{(m-1)/2,2}\sin 2\pi/3, \\ a_{m-1,2} &= a_{(m-1)/2,1}\sin 2\pi/3 + a_{(m-1)/2,2}\cos 2\pi/3, \\ a_{m,1} &= a_{(m-1)/2,1}\cos 4\pi/3 - a_{(m-1)/2,2}\sin 4\pi/3, \\ a_{m,2} &= a_{(m-1)/2,1}\sin 4\pi/3 + a_{(m-1)/2,2}\cos 4\pi/3. \end{aligned}$$

For the design factor $n = 50$, only the first 50 rows of $\mathbf{H}$ were used, and for $n = 100$ all rows were used. To construct the dataset $\mathbf{Z}$, an error matrix $\mathbf{E}$ with independent elements $e_{ij} \sim N(0, 1)$ was added using

$$\mathbf{Z} = (1 - \varepsilon)^{1/2}\mathbf{HA}' + \varepsilon^{1/2}\mathbf{E},$$

where ε specifies the error level with $0 < \varepsilon < 1$. All cells in the design used the same 150 two-dimensional structures $\mathbf{H}$, hereby further reducing the variability due to other factors than the design factors. The error $\mathbf{E}$ was different in every replication in all cells of the design.

References

Borg, I., & Groenen, P.J.F. (1997). *Modern multidimensional scaling: Theory and applications*. New York: Springer.

Breiman, L., & Friedman, J.H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, *80*, 580–619.

Bro, R., & De Jong, S. (1997). A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, *11*, 393–401.

De Boor, C. (1978). *A practical guide to splines*. New York: Springer.

de Leeuw, J. (1977). *Normalized cone regression*. Leiden, The Netherlands: Department of Data Theory.

de Leeuw, J., & Heiser, W.J. (1977). Convergence of correction-matrix algorithms for multidimensional scaling. In J.C. Lingoes, E.E. Roskam, & I. Borg (Eds.), *Geometric representations of relational data* (pp. 735–752). Ann Arbor, MI: Mathesis Press.

Dinkelbach, W. (1967). On nonlinear fractional programming. *Managment Science*, *13*, 492–498.

Gifi, A. (1990). *Nonlinear multivariate analysis*. Chichester: Wiley.

Gower, J.C., & Hand, D.J. (1996). *Biplots*. London: Chapman and Hall.

Groenen, P.J.F., Commandeur, J.J.F., & Meulman, J.J. (1997). PIONEER: A program for distance-based multivariate analysis. In W. Bandilla & F. Faulbaum (Eds.), *Softstat'97: Advances in statistical software* (pp. 83–90). Stuttgart: Lucius & Lucius.

Heiser, W.J. (1985). *Multidimensional scaling by optimizing goodness-of-fit to a smooth hypothesis* (Tech. Rep. RR-85-07). Leiden, The Netherlands: Leiden University, Department of Data Theory.

Jones, L.K., & Seaton, L. (Eds.). (1994). *Math77: Reference manual*. Sterling, VA: Language Systems Coorporation.

Kiers, H. A.L. (1995). Maximization of sums of quotients of quadratic forms and some generalizations. *Psychometrika*, *60*, 221–245.

Kruskal, J.B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*, 1–27.

Kruskal, J.B. (1964b). Nonmetric multidimensional scaling: A numerical method, *Psychometrika*, *29*, 115–129.

Kruskal, J.B., & Carroll, J.D. (1969). Geometrical models and badness-of-fit functions. In P.R. Krishnaiah (Ed.), *Multivariate analysis* (Vol. 2, pp. 639–671). New York: Academic Press.

Krzanowski, W.J., & Marriott, F.H.C. (1994). *Multivariate analysis. Part 1: Distributions, ordination and inference*. London: Wiley.

Lawson, C.L., & Hanson, R.J. (1974). *Solving least squares problems*. Englewood Cliffs, NJ: Prentice Hall.

Meulman, J.J. (1986). *A distance approach to nonlinear multivariate analysis*. Leiden, The Netherlands: DSWO Press.

Meulman, J.J. (1992). The integration of multidimensional scaling and multivariate analysis with optimal transformations. *Psychometrika*, *57*, 539–565.

Meyer, R. (1997). Nonlinear eigenvector algorithms for local optimization in multivariate data analysis. *Linear Algebra and its Applications*, *264*, 225–246.

Ramsay, J.O. (1988). Monotone regression splines in action. *Statistical Science*, *3*, 425–461.

Rockafellar, R.T. (1970). *Convex analysis*. Princeton, NJ: Princeton University Press.

Verboon, P. (1994). *A robust approach to nonlinear multivariate analysis*. Leiden, The Netherlands: DSWO Press.

Watson, G.A. (1985). On a class of algorithms for total approximation. *Journal of Approximation Theory*, *45*, 219–231.

Young, F.W., de Leeuw, J., & Takane, Y. (1976). Regression with qualitative and quantitative variables: An alternating least squares approach. *Psychometrika*, *41*, 505–529.

Manuscript received 9 FEB 1998
Final version received 16 OCT 1999