PSYCHOMETRIKA—VOL. 60, NO. 2, 221–245
JUNE 1995

MAXIMIZATION OF SUMS OF QUOTIENTS OF QUADRATIC FORMS AND SOME GENERALIZATIONS

HENK A. L. KIERS

UNIVERSITY OF GRONINGEN

Monotonically convergent algorithms are described for maximizing six (constrained) functions of vectors $\mathbf{x}$, or matrices X with columns $\mathbf{x}_1, \ldots, \mathbf{x}_r$. These functions are $\mathrm{h}_1(\mathbf{x}) = \sum_k (\mathbf{x}'A_k\mathbf{x})(\mathbf{x}'C_k\mathbf{x})^{-1}$, $\mathrm{H}_1(X) = \sum_k \mathrm{tr}\,(X'A_kX)(X'C_kX)^{-1}$, $\tilde{\mathrm{h}}_1(X) = \sum_k \sum_l (\mathbf{x}_l'A_k\mathbf{x}_l)(\mathbf{x}_l'C_k\mathbf{x}_l)^{-1}$ with X constrained to be columnwise orthonormal, $\mathrm{h}_2(\mathbf{x}) = \sum_k (\mathbf{x}'A_k\mathbf{x})^2(\mathbf{x}'C_k\mathbf{x})^{-1}$ subject to $\mathbf{x}'\mathbf{x} = 1$, $\mathrm{H}_2(X) = \sum_k \mathrm{tr}\,(X'A_kX)(X'A_kX)'(X'C_kX)^{-1}$ subject to $X'X = I$, and $\tilde{\mathrm{h}}_2(X) = \sum_k \sum_l (\mathbf{x}_l'A_k\mathbf{x}_l)^2(\mathbf{x}_l'C_k\mathbf{x}_l)^{-1}$ subject to $X'X = I$. In these functions the matrices C_k are assumed to be positive definite. The matrices A_k can be arbitrary square matrices. The general formulation of the functions and the algorithms allows for application of the algorithms in various problems that arise in multivariate analysis. Several applications of the general algorithms are given. Specifically, algorithms are given for reciprocal principal components analysis, binormamin rotation, generalized discriminant analysis, variants of generalized principal components analysis, simple structure rotation for one of the latter variants, and set component analysis. For most of these methods the algorithms appear to be new, for the others the existing algorithms turn out to be special cases of the newly derived general algorithms.

Key words: generalized principal components analysis, generalized discriminant analysis, binormamin, simple structure rotation.

Several techniques for multivariate data analysis involve the optimization of a quotient of two quadratic forms. A well-known example is discriminant analysis, where the central problem is to maximize

$$\mathrm{f}(\mathbf{x}) = \frac{\mathbf{x}'B\mathbf{x}}{\mathbf{x}'W\mathbf{x}}, \tag{1}$$

over $\mathbf{x}$, where B is the matrix of between groups covariances and W is the matrix of within groups covariances (see, e.g., Tatsuoka 1971, p. 159).

The problem of maximizing $\mathrm{f}(\mathbf{x})$ has a straightforward solution, when W is positive definite (p.d.). This solution is obtained by first reparameterizing $\mathbf{y} \equiv W^{1/2}\mathbf{x}$, and next solving the equivalent but simpler problem of maximizing $\mathrm{g}(\mathbf{y}) = (\mathbf{y}'W^{-1/2}BW^{-1/2}\mathbf{y})(\mathbf{y}'\mathbf{y})^{-1}$ over $\mathbf{y}$. The latter problem is solved by taking $\mathbf{y}$ equal to the first eigenvector of $W^{-1/2}BW^{-1/2}$, from which $\mathbf{x}$ follows as $\mathbf{x} = W^{-1/2}\mathbf{y}$. As an aside, it might be mentioned that the problem has been considered in a wider context by McDonald (1968). Furthermore, solutions for the case where W is singular have been developed by McDonald, Torii and Nishisato (1979) and de Leeuw (1982).

In a number of methods, the aim is to maximize a generalization of the function $\mathrm{f}(\mathbf{x})$. Some such generalizations have the form

This research has been made possible by a fellowship from the Royal Netherlands Academy of Arts and Sciences to the author. The author is obliged to Jos ten Berge for stimulating this research and for helpful comments on an earlier version of this paper.

Requests for reprints should be sent to Henk A. L. Kiers, Department of Psychology (SPA), University of Groningen, Grote Kruisstraat 2/1, 9712 TS Groningen, THE NETHERLANDS.

0033-3123/95/0600-93115$00.75/0

$$\mathrm{h}_1(\mathbf{x}) = \sum_{k=1}^{p} \frac{\mathbf{x}'A_k\mathbf{x}}{\mathbf{x}'C_k\mathbf{x}}, \tag{2}$$

where A_k and C_k are $n \times n$ matrices and C_k is p.d., $k = 1, \ldots, p$. If all C_k are equal, we can replace them by $W \equiv C_k$, and we have $\mathrm{h}_1(\mathbf{x}) = \mathbf{x}' \sum_k A_k\mathbf{x}/\mathbf{x}'W\mathbf{x}$. Hence, in this case $\mathrm{h}_1(\mathbf{x})$ is a function of the form of f(x), and the maximum is obtained as described above.

Unfortunately, in cases where the matrices C_k differ, no such simple solution is available. It is the purpose of the present paper to propose iterative algorithms for maximizing $\mathrm{h}_1(\mathbf{x})$ and similar functions consisting of sums of quotients of quadratic forms. Specifically, apart from the maximization of $\mathrm{h}_1(\mathbf{x})$, we will discuss the maximization of

$$\mathrm{H}_1(X) = \sum_{k=1}^{p} \mathrm{tr}\, (X'A_kX)(X'C_kX)^{-1}, \tag{3}$$

where X is an arbitrary $n \times r$ $(r < n)$ matrix that has full column rank (r), and of

$$\tilde{\mathrm{h}}_1(X) = \sum_{k=1}^{p} \sum_{l=1}^{r} \frac{\mathbf{x}_l'A_k\mathbf{x}_l}{\mathbf{x}_l'C_k\mathbf{x}_l}, \tag{4}$$

where $\mathbf{x}_l$, $l = 1, \ldots, r$, denotes the columns of X. Without constraint on X, the X maximizing $\tilde{\mathrm{h}}_1(X)$ would consist of r equal columns, all maximizing h_1. To avoid such degenerate solutions, we impose the constraint that X is columnwise orthonormal. In fact, we can impose this constraint on $\mathrm{H}_1(X)$ as well, without affecting the fit, as is readily verified. Similarly, we can impose the identification constraint $\mathbf{x}'\mathbf{x} = 1$. It can now be seen that $\mathrm{H}_1(X)$ and $\tilde{\mathrm{h}}_1(X)$ both reduce to $\mathrm{h}_1(\mathbf{x})$ in case X has only one column.

In addition to the three problems mentioned above, we will discuss the maximization of the functions

$$\mathrm{h}_2(\mathbf{x}) = \sum_{k=1}^{p} \frac{(\mathbf{x}'A_k\mathbf{x})^2}{\mathbf{x}'C_k\mathbf{x}} \tag{5}$$

subject to $\mathbf{x}'\mathbf{x} = 1$, that of

$$\mathrm{H}_2(X) = \sum_{k=1}^{p} \mathrm{tr}\, (X'A_kX)(X'A_k'X)(X'C_kX)^{-1} \tag{6}$$

over X subject to $X'X = I$, and that of

$$\tilde{\mathrm{h}}_2(X) = \sum_{k=1}^{p} \sum_{l=1}^{r} \frac{(\mathbf{x}_l'A_k\mathbf{x}_l)^2}{\mathbf{x}_l'C_k\mathbf{x}_l} \tag{7}$$

also subject to $X'X = I$. In these three problems, C_k is again assumed to be p.d., and A_k can be any square matrix. All problems are summarized in Table 1.

Most of the above problems have in some form or an other appeared in the literature. For instance, Kaiser and Dickman's (1959) Binormamin method for oblique simple structure rotation implies maximization of functions of the form $\mathrm{h}_1(\mathbf{x})$. Kiers and

TABLE 1

Summary of the Six Maximization Problems

Function Maximized	Constraint
$h_1(\mathbf{x}) = \sum_{k=1}^{p} \frac{\mathbf{x}'A_k\mathbf{x}}{\mathbf{x}'C_k\mathbf{x}}$,	none ($\mathbf{x}'\mathbf{x}=1$ for identification)
$H_1(X) = \sum_{k=1}^{p} \mathrm{tr}(X'A_kX)(X'C_kX)^{-1}$,	none ($X'X = I$ for identification)
$\tilde{h}_1(X) = \sum_{k=1}^{p} \sum_{l=1}^{r} \frac{\mathbf{x}_l'A_k\mathbf{x}_l}{\mathbf{x}_l'C_k\mathbf{x}_l}$,	$X'X = I$
$h_2(\mathbf{x}) = \sum_{k=1}^{p} \frac{(\mathbf{x}'A_k\mathbf{x})^2}{\mathbf{x}'C_k\mathbf{x}}$	$X'X = I$
$H_2(X) = \sum_{k=1}^{p} \mathrm{tr}(X'A_kX)(X'A_k'X)(X'C_kX)^{-1}$	$X'X = I$
$\tilde{h}_2(X) = \sum_{k=1}^{p} \sum_{l=1}^{r} \frac{(\mathbf{x}_l'A_k\mathbf{x}_l)^2}{\mathbf{x}_l'C_k\mathbf{x}_l}$	$X'X = I$

ten Berge (1994) have proposed a method for rotation to simple structure that involves the maximization of functions of the form $h_1(\mathbf{x})$. Millsap and Meredith's (1988) methods for component analysis in cross-sectional and longitudinal data employ the maximization of a function of the form $H_1(X)$, and so do Kiers and ten Berge's (1994) methods for simultaneous components analysis. Finally, Nierop (1993) has proposed a method for set components analysis which essentially maximizes a function of the form $\tilde{h}_2(X)$, or its special case $h_2(\mathbf{x})$. In the present paper, we will give algorithms for all these methods, as well as for some new methods (e.g., a straightforward generalization of discriminant analysis), based on the algorithms for the general functions (2) through (7).

All algorithms to be proposed here are iterative algorithms that update X (subject to the constraints at hand) such that the function increases monotonically. To derive these algorithms, we will first consider the maximization of a general function that has H_1 and H_2 as special cases (and hence h_1 and h_2, as well). In the next section, it will be shown that this function can be increased by the X that minimizes (or decreases) an auxiliary function. In the subsequent sections, it will be shown how H_1, h_1 and $\tilde{h}_1$ can be maximized by decreasing the respective auxiliary functions associated with each of these functions. The algorithms for maximizing H_1, h_1 and $\tilde{h}_1$ are based on the assumption that A_k is positive semidefinite (p.s.d.). In a subsequent section, it will be shown how the algorithms can be modified to handle cases where A_k, $k = 1, \ldots, p$, is not p.s.d. As an aside, it is shown how the functions H_1, h_1, and $\tilde{h}_1$ can be *minimized*. Next, the constrained maximization of H_2, h_2, and $\tilde{h}_2$ will be discussed. Then, some results on convergence properties and the performance of all algorithms will be reported. Finally, several applications of the algorithms are discussed. For easy reference, a schematic overview of the algorithms is given in Appendix A.

Increasing H_1 or H_2 by Decreasing an Auxiliary Function

The algorithms for maximizing H_1 and H_2 to be derived below, will be based on decreasing an auxiliary function. The algorithms for maximizing h_1, $\tilde{h}_1$, h_2 and $\tilde{h}_2$ will be based on straightforward variants of this auxiliary function. The auxiliary function can be found for H_1 and H_2 simultaneously when we consider *increasing* H_1 and H_2 as special cases of *decreasing* the general function

$$H(X) = -\sum_{k=1}^{p} \operatorname{tr} (X'C_kX)^{-s}F_k(X)F_k(X)', \tag{8}$$

where $F_k(X)$ is a matrix function of X. When $s = 1$ and $F_k(X) = X'A_k^{1/2}$ (assuming that A_k is p.s.d.), we obtain $H(X) = -H_1(X)$; $s = 1$ and $F_k(X) = X'A_kX$ yields $H(X) = -H_2(X)$. Therefore, a procedure that decreases $H(X)$ can be used to increase both H_1 and H_2. It remains to find such a procedure. This will be based on "majorization" (see, for instance, de Leeuw & Heiser, 1980; Kiers, 1990; Meulman, 1986).

Let the current X be denoted by X_0. Then, to decrease H, we have to find a matrix X such that $H(X) \le H(X_0)$. Suppose we have a function $G(X)$ such that $H(X) \le G(X)$ for all X (i.e., $G(X)$ majorizes $H(X)$), and $H(X_0) = G(X_0)$. Also, suppose that we have a procedure for updating X_0 by X_m such that $G(X_m) \le G(X_0)$. Then this procedure decreases H, because $H(X_m) \le G(X_m) \le G(X_0) = H(X_0)$. Hence, if we can find a function G such that $H(X) \le G(X)$ for all X (subject to the constraint at hand), and $H(X_0) = G(X_0)$, and if we know how to minimize (or at least decrease) G, we can construct an algorithm that monotonically decreases $H(X)$. Such a function $G(X)$ can be derived as follows.

From the inequality

$$\|(X'C_kX)^{-\frac{1}{2}s}F_k(X) - (X'C_kX)^{\frac{1}{2}s}(X_0'C_kX_0)^{-s}F_k(X_0)\|^2 \ge 0, \tag{9}$$

where it is used that X_0 has full column rank, it follows that

$$\operatorname{tr} (X'C_kX)^{-s}F_k(X)F_k(X)' \ge 2 \operatorname{tr} (X_0'C_kX_0)^{-s}F_k(X_0)F_k(X)' - \operatorname{tr} (X'C_kX)^{s}(X_0'C_kX_0)^{-s}F_k(X_0)F_k(X_0)'(X_0'C_kX_0)^{-s}. \tag{10}$$

Defining $W_k \equiv F_k(X_0)'(X_0'C_kX_0)^{-s}$, which is independent of X, we obtain, after summing over k, and multiplying by -1,

$$H(X) = -\sum_{k=1}^{p} \operatorname{tr} (X'C_kX)^{-s}F_k(X)F_k(X)'$$

$$\le -2 \sum_{k=1}^{p} \operatorname{tr} W_k'F_k(X)' + \sum_{k=1}^{p} \operatorname{tr} W_k'W_k(X'C_kX)^{s} \equiv G(X). \tag{11}$$

It is important to note that, for the thus defined function $G(X)$, we have $G(X_0) = H(X_0)$. In fact, the function $G(X)$ depends on X_0, because W_k is different for different values of X_0. For that reason it would have been better to denote it as $G_{X_0}(X)$, but to enhance readability, it has been chosen to maintain the simplified notation $G(X)$. From the above reasoning, it follows that by minimizing or decreasing $G(X)$ we can decrease $H(X)$. For $F_k(X) = X'A_k^{1/2}$ we denote $G(X)$ as $G_1(X)$, hence

$$\mathrm{G}_1(X) = -2 \sum_{k=1}^{p} \operatorname{tr} W_k' \underline{A_k^{1/2} X} + \sum_{k=1}^{p} \operatorname{tr} W_k' W_k X' C_k X, \tag{12}$$

with $W_k = A_k^{1/2} X_0 (X_0' C_k X_0)^{-1}$; for $\mathrm{F}_k(X) = X' A_k X$ we denote $\mathrm{G}(X)$ as $\mathrm{G}_2(X)$, hence

$$\mathrm{G}_2(X) = -2 \sum_{k=1}^{p} \operatorname{tr} X' A_k X W_k + \sum_{k=1}^{p} \operatorname{tr} W_k' W_k X' C_k X, \tag{13}$$

with $W_k = X_0' A_k' X_0 (X_0' C_k X_0)^{-1}$.

In the next section, it will be shown how $\mathrm{H}_1(X)$, with A_k assumed to be p.s.d., $k = 1, \ldots, p$, can be increased by minimizing $\mathrm{G}_1(X)$. The maximization of $\mathrm{h}_1(\mathbf{x})$ will be considered as a special case of that of $\mathrm{H}_1(X)$. The maximization of $\bar{\mathrm{h}}_1(X)$ will be treated by decreasing, rather than minimizing, an auxiliary function based on $\mathrm{G}_1(X)$.

Algorithms for Maximizing $\mathrm{H}_1(X)$, $\mathrm{h}_1(\mathbf{x})$ and $\bar{\mathrm{h}}_1(X)$

In the previous section, it has been described that we can increase $\mathrm{H}_1(X)$ by minimizing the function $\mathrm{G}_1(X)$ defined in (12), provided that A_k be p.s.d., $k = 1, \ldots, p$. In the next section, it will be shown how it can always be arranged that A_k is p.s.d. In the present section, it will be shown how $\mathrm{G}_1(X)$ can be minimized, and how this leads to an algorithm for maximizing $\mathrm{H}_1(X)$ and its special case $\mathrm{h}_1(\mathbf{x})$. Along similar lines, an algorithm will be derived for maximizing $\bar{\mathrm{h}}_1(X)$. To facilitate programming the algorithms, all algorithms are summarized schematically in Appendix A.

An Algorithm for Maximizing $\mathrm{H}_1(X)$

As has been shown in the previous section, the function $\mathrm{G}_1(X)$ majorizes $-\mathrm{H}_1(X) = -\sum_k \operatorname{tr} (X' A_k X)(X' C_k X)^{-1}$. To see how $\mathrm{G}_1(X)$ can be minimized, we express it as

$$\begin{aligned}
\mathrm{G}_1(X) &= \sum_{k=1}^{p} \|C_k^{-1/2} A_k^{1/2} - C_k^{1/2} X W_k'\|^2 - \sum_{k=1}^{p} \|C_k^{-1/2} A_k^{1/2}\|^2 \\
&= \sum_{k=1}^{p} \| \operatorname{Vec} (C_k^{-1/2} A_k^{1/2}) - (W_k \otimes C_k^{1/2}) \operatorname{Vec} (X)\|^2 - \sum_{k=1}^{p} \|C_k^{-1/2} A_k^{1/2}\|^2 \\
&= \left\| \begin{pmatrix} \operatorname{Vec} (C_1^{-1/2} A_1^{1/2}) \\ \vdots \\ \operatorname{Vec} (C_p^{-1/2} A_p^{1/2}) \end{pmatrix} - \begin{pmatrix} W_1 \otimes C_1^{1/2} \\ \vdots \\ W_p \otimes C_p^{1/2} \end{pmatrix} \operatorname{Vec} (X) \right\|^2 - \sum_{k=1}^{p} \|C_k^{-1/2} A_k^{1/2}\|^2,
\end{aligned} \tag{14}$$

where $\otimes$ denotes the Kronecker product, and Vec denotes a matrix strung out columnwise into a vector; the second equality is obtained from the relation $\operatorname{Vec}(ABC) = (C' \otimes A)\operatorname{Vec}(B)$ (see, e.g., Henderson & Searle, 1981, p. 273). Finding the minimum of (14) over X is a regression problem (where $\operatorname{Vec}(X)$ contains the regression weights) and is solved by

$$\text{Vec}(X) = \left(\sum_{k=1}^{p} (W_k' W_k) \otimes C_k\right)^{-1} \sum_{k=1}^{p} (W_k' \otimes C_k^{1/2}) \text{ Vec } (C_k^{-1/2} A_k^{1/2})$$

$$= \left(\sum_{k=1}^{p} (W_k' W_k) \otimes C_k\right)^{-1} \sum_{k=1}^{p} \text{Vec } (C_k^{1/2} C_k^{-1/2} A_k^{1/2} W_k)$$

$$= \left(\sum_{k=1}^{p} (W_k' W_k) \otimes C_k\right)^{-1} \sum_{k=1}^{p} \text{Vec } (A_k^{1/2} W_k), \quad (15)$$

assuming that inverses exist, or else using generalized inverses. From expression (15) for Vec (X), we obtain the X that minimizes $\mathrm{G}_1(X)$ by taking as its columns the subcolumns (of n elements) of Vec(X). Thus, we have a procedure for minimizing $\mathrm{G}_1(X)$ and hence for increasing $\mathrm{H}_1(X)$ monotonically. If H_1 is bounded above, this procedure must converge to a stable function value of H_1. Indeed, the function H_1 is bounded above, as can be seen as follows. For each term of H_1 we have $\text{tr}\,(X'C_kX)^{-1}(X'A_kX) = \text{tr}\, Y_k' C_k^{-1/2} A_k C_k^{-1/2} Y_k$, where $Y_k \equiv C_k^{1/2} X (X' C_k X)^{-1/2}$. Because Y_k is columnwise orthonormal, $\text{tr}\, Y_k' C_k^{-1/2} \overline{A_k C_k}^{-1/2} Y_k$ is smaller than or equal to the sum of the first r eigenvalues of $C_k^{-1/2} A_k C_k^{-1/2}$.

As has already been mentioned, we can impose $X'X = I$ without loss of generality. This is because orthonormalizing X does not affect the function value (as is readily verified). If desired, one can even orthonormalize X during the iterative process, immediately after updating X.

It has been mentioned in the introduction that X should have full column rank (r), because otherwise H_1 is not defined. In the above algorithm this constraint has not been effectuated, and, in fact, it is conceivable that a low rank matrix X is found during the iterations. However, this is very unlikely to happen in practice.

An Algorithm for Maximizing $\mathrm{h}_1(\mathbf{x})$: *The Special Case Where X is a Vector*

In the special case where X has only one column, $\mathrm{H}_1(X)$ reduces to $\mathrm{h}_1(\mathbf{x}) = \sum_k \text{tr}\,(\mathbf{x}'A_k\mathbf{x})(\mathbf{x}'C_k\mathbf{x})^{-1}$, W_k reduces to $\mathbf{w}_k = (\mathbf{x}_0'C_k\mathbf{x}_0)^{-1}A_k^{1/2}\mathbf{x}_0$, and the update for $\mathbf{x}$ can be derived from (15) as

$$\mathbf{x} = \left(\sum_{k=1}^{p} \mathbf{w}_k'\mathbf{w}_k C_k\right)^{-1} \sum_{k=1}^{p} (A_k^{1/2}\mathbf{w}_k)$$

$$= \left(\sum_{k=1}^{p} (\mathbf{x}_0'C_k\mathbf{x}_0)^{-2}(\mathbf{x}_0'A_k\mathbf{x}_0)C_k\right)^{-1} \sum_{k=1}^{p} ((\mathbf{x}_0'C_k\mathbf{x}_0)^{-1}A_k)\mathbf{x}_0. \quad (16)$$

Hence, iteratively updating $\mathbf{x}$ according to (16) increases $\mathrm{h}_1(\mathbf{x})$ monotonically and converges to a stable function value of $\mathrm{h}_1(\mathbf{x})$. Again, if desired, $\mathbf{x}$ can be normalized to unit sum of squares, either after each update, or after convergence.

An alternative, but, as we will see, unreliable, procedure for obtaining an algorithm to maximize $\mathrm{h}_1(\mathbf{x})$ is based on the normal equation for h_1. Taking the derivative of h_1 with respect to $\mathbf{x}$, and setting this equal to zero we obtain as a necessary condition for the maximum of $\mathrm{h}_1(\mathbf{x})$ that

$$\sum_{k=1}^{p}\left(\frac{1}{\mathbf{x}'C_k\mathbf{x}}A_k\right)\mathbf{x}=\sum_{k=1}^{p}\left(\frac{\mathbf{x}'A_k\mathbf{x}}{(\mathbf{x}'C_k\mathbf{x})^2}C_k\right)\mathbf{x}. \tag{17}$$

There are two simple ways of "explicitating" $\mathbf{x}$ in this implicit equation for $\mathbf{x}$. The first is to express $\mathbf{x}$ as

$$\mathbf{x}=\left(\sum_{k=1}^{p}\frac{\mathbf{x}'A_k\mathbf{x}}{(\mathbf{x}'C_k\mathbf{x})^2}C_k\right)^{-1}\left(\sum_{k=1}^{p}\frac{1}{\mathbf{x}'C_k\mathbf{x}}A_k\right)\mathbf{x}. \tag{18}$$

Because (18) must hold at every stationary point of h_1, one might conjecture that an iterative procedure in which one computes a new $\mathbf{x}$ according to the right-hand side of (18), iteratively, will converge to a stable solution for $\mathbf{x}$ that satisfies (18), and, hopefully gives the $\mathbf{x}$ that maximizes h_1, or at least finds a local maximum. As a matter of fact, this procedure for updating $\mathbf{x}$ is the very same as the algorithm, based on (16), for which we have proven monotone convergence above. Hence, indeed the conjecture that such an algorithm can be used to find local maxima holds true. However, this is just a matter of coincidence, because, rather than "explicitating" $\mathbf{x}$ as in (18), we might just as well explicitate $\mathbf{x}$ as

$$\mathbf{x}=\left(\sum_{k=1}^{p}\frac{1}{\mathbf{x}'C_k\mathbf{x}}A_k\right)^{-1}\left(\sum_{k=1}^{p}\frac{\mathbf{x}'A_k\mathbf{x}}{(\mathbf{x}'C_k\mathbf{x})^2}C_k\right)\mathbf{x}. \tag{19}$$

This expression suggests an algorithm in which $\mathbf{x}$ is updated iteratively according to the right-hand side of (19). This algorithm has been programmed and it was found that it neither increased function h_1 monotonically, nor always converged. In fact, a (random) data set was found for which the algorithm oscillated between two different function values. It can be concluded that building an algorithm on the basis of the normal equation alone is a hazardous enterprise, which may lead to nonconvergent algorithms.

An Algorithm for Maximizing $\bar{\mathrm{h}}_1(X)$

The maximization of the function $\bar{\mathrm{h}}_1(X)=\sum_l \mathrm{h}_1(\mathbf{x}_l)=\sum_k\sum_l \operatorname{tr}(\mathbf{x}_l'A_k\mathbf{x}_l)(\mathbf{x}_l'C_k\mathbf{x}_l)^{-1}$, subject to $X'X=I$, can be handled via the minimization of a majorizing function, as follows. Because $\bar{\mathrm{h}}_1(X)$ is not a special case of $\mathrm{H}_1(X)$, we cannot use the minimization of $\mathrm{G}_1(X)$ for that purpose. However, it is easy to derive a majorizing function, by summing functions that majorize $-\mathrm{h}_1$. Specifically, the function $-\bar{\mathrm{h}}_1(X)$ is majorized by

$$\bar{\mathrm{g}}_1(X)\equiv\sum_{l=1}^{r}\mathrm{G}_1(\mathbf{x}_l)=-2\sum_{k=1}^{p}\sum_{l=1}^{r}\mathbf{w}_{kl}'A_k^{1/2}\mathbf{x}_l+\sum_{k=1}^{p}\sum_{l=1}^{r}\mathbf{w}_{kl}'\mathbf{w}_{kl}\mathbf{x}_l'C_k\mathbf{x}_l, \tag{20}$$

where $\mathbf{w}_{kl}\equiv(\mathbf{x}_{l0}'C_k\mathbf{x}_{l0})^{-1}A_k^{1/2}\mathbf{x}_{l0}$, and $\mathbf{x}_{l0}$ denotes the current $\mathbf{x}_l$. It is readily verified that $\bar{\mathrm{g}}_1(X_0)=\bar{\mathrm{h}}_1(X_0)$. Hence, we can decrease $\bar{\mathrm{h}}_1$ by iteratively minimizing or, which is simpler here, decreasing $\bar{\mathrm{g}}_1$. To see how we can decrease $\bar{\mathrm{g}}_1$, we first rewrite it as follows. Let W_k be defined as the $n\times r$ matrix with columns $\mathbf{w}_{kl}$, and D_k as the $r\times r$ diagonal matrix with elements $\mathbf{w}_{kl}'\mathbf{w}_{kl}$ on the diagonal, $l=1,\ldots,r$, $k=1,\ldots,p$, hence $D_k=(\operatorname{Diag}(X_0'A_kX_0))(\operatorname{Diag}(X_0'C_kX_0))^{-2}$. Then we can rewrite $\bar{\mathrm{g}}_1(X)$ as

$$\bar{g}_1(X) = -2 \sum_{k=1}^{p} \text{tr } W_k' A_k^{1/2} X + \sum_{k=1}^{p} \text{tr } X' C_k X D_k. \tag{21}$$

The problem of decreasing $\bar{g}_1(X)$ over columnwise orthonormal X can be handled by a procedure described by Kiers and ten Berge (1992). Specifically, they have proved that we decrease $\bar{g}_1$ by taking X as PQ', where P and Q are taken from the singular value decomposition (SVD)

$$M = 2 \sum_{k=1}^{p} \rho_k X_0 D_k - 2 \sum_{k=1}^{p} C_k X_0 D_k + 2 \sum_{k=1}^{p} A_k^{1/2} W_k = PDQ', \tag{22}$$

and ρ_k is taken larger than or equal to the first eigenvalue of C_k (see Kiers & ten Berge, pp. 374–375). Note that in (22) we use $A_k^{1/2}$, but in the algorithm we do not need to compute $A_k^{1/2}$ explicitly, because the term in which it appears, $A_k^{1/2} W_k$, equals $A_k X_0(\text{Diag }(X_0' C_k X_0))^{-1}$, and thus involves A_k itself rather than its square root. On the basis of (22), one can construct an algorithm for iteratively updating X such that $\bar{h}_1(X)$ increases monotonically, and, because $\bar{h}_1(X)$ is bounded above, the algorithm must converge to a stable function value.

It is interesting to note that, in the special case where $C_k = I$, $k = 1, \ldots, p$, we have $\bar{h}_1(X) = \text{tr } X' \sum_k A_k X$, and the algorithm reduces to computing $M = 2 \sum_k X_0 D_k - 2 \sum_k X_0 D_k + 2 \sum_k A_k^{1/2} W_k = 2 \sum_k A_k X_0(\text{Diag }(X_0' X_0))^{-1} = 2 \sum_k A_k X_0$, obtaining its SVD $M = PDQ'$, and setting $X = PQ'$, iteratively. This is equivalent to the Bauer-Rutishauser algorithm (see Kroonenberg & de Leeuw, 1980, pp. 74–75) applied to $\sum_k A_k$, a generalization of the power method for computing the first r eigenvectors of a symmetric matrix.

In the present section, it has been assumed throughout that A_k is p.s.d., $k = 1, \ldots, p$. In the next section, it will be shown how we can modify the algorithms such that they can be used for maximization in cases where A_k is not p.s.d. Moreover, it will be shown how the procedures can even be modified to *minimize* the functions at hand.

Maximizing and Minimizing $\text{H}_1(X)$, $\text{h}_1(\mathbf{x})$ and $\bar{h}_1(X)$, when A_k is not p.s.d.

The assumption that A_k is p.s.d., $k = 1, \ldots, p$, was crucial in the derivations of the algorithms in the previous section. Nevertheless, it is fairly simple to find algorithms for the maximization of these functions when A_k is not p.s.d., as will be shown in the present section. As a by-product, it will also be shown how we can minimize rather than maximize these functions. First, the case where A_k is not p.s.d. but merely symmetric will be treated. Next, we turn to the case where A_k is not even symmetric (but merely square). Finally, we will consider the *minimization* of the functions at hand.

Case 1: A_k Symmetric but not p.s.d.

We can rewrite $\text{H}_1(X)$ as

$$\text{H}_1(X) = \sum_{k=1}^{p} \text{tr } (X'(A_k - \alpha_k C_k)X)(X' C_k X)^{-1} + r \sum_{k=1}^{p} \alpha_k, \tag{23}$$

for arbitrary scalars α_k, $k = 1, \ldots, p$. We choose α_k such that $\tilde{A}_k \equiv (A_k - \alpha_k C_k)$ is p.s.d., $k = 1, \ldots, p$. The second term of $\text{H}_1(X)$ is constant and the first is of the same form as $\text{H}_1(X)$ but now involving p.s.d. matrices $\tilde{A}_k$. Hence, the algorithm derived in the previous section can, after replacement of A_k by $\tilde{A}_k$, be used to maximize $\text{H}_1(X)$. Obviously, for the maximization of $\text{h}_1(\mathbf{x})$ and $\tilde{\text{h}}_1(X)$ the same replacement can be used.

It remains to find out how to choose α_k such that $A_k - \alpha_k C_k$ is p.s.d. As above, it is assumed that C_k is p.d., hence we can write $(A_k - \alpha_k C_k) = C_k^{1/2} C_k^{-1/2}(A_k - \alpha_k C_k) C_k^{-1/2} C_k^{1/2}$, which is p.s.d. as soon as $C_k^{-1/2}(A_k - \alpha_k C_k)C_k^{-1/2} = (C_k^{-1/2} A_k C_k^{-1/2} - \alpha_k I)$ is p.s.d. The latter matrix is p.s.d. if we choose α_k smaller than or equal to the smallest eigenvalue of $C_k^{-1/2} A_k C_k^{-1/2}$. Clearly, if A_k is p.s.d. we can take α_k equal to 0.

Case 2: A_k Asymmetric

If A_k is asymmetric, we first write A_k as $(S_k + N_k)$, where $S_k = \frac{1}{2}(A_k + A'_k)$, called the "symmetric part" of A_k, and $N_k = \frac{1}{2}(A_k - A'_k)$, called the "skew-symmetric part" of A_k. Then $\text{H}_1(X)$ can be rewritten as

$$\text{H}_1(X) = \sum_{k=1}^{p} \text{tr}\,(X'A_kX)(X'C_kX)^{-1} = \sum_{k=1}^{p} \text{tr}\,(X'(S_k + N_k)X)(X'C_kX)^{-1}$$

$$= \sum_{k=1}^{p} \text{tr}\,(X'S_kX)(X'C_kX)^{-1} + \sum_{k=1}^{p} \text{tr}\,(X'N_kX)(X'C_kX)^{-1}. \tag{24}$$

As is readily verified, the trace of the product of a symmetric and a skew-symmetric matrix is zero. Hence, the second term in the right-hand side of (24) vanishes. Therefore, without affecting the function value, we can replace A_k by S_k in $\text{H}_1(X)$, and, obviously, also in $\text{h}_1(\mathbf{x})$ and $\tilde{\text{h}}_1(X)$. If S_k is p.s.d., we can proceed as with the original H_1, h_1 and $\tilde{\text{h}}_1$ functions. If S_k is not p.(s.)d., we can replace it by $S_k - \alpha_k C_k$ to make it p.s.d., as in Case 1.

Case 3: Minimizing H_1, h_1 *and* $\tilde{\text{h}}_1$

Minimizing $\text{H}_1(X)$, $\text{h}_1(\mathbf{x})$ or $\tilde{\text{h}}_1(X)$ is equivalent to maximizing the same function with A_k replaced by $-A_k$, $k = 1, \ldots, p$. If $-A_k$ is p.s.d. we can proceed as in the previous section. If $-A_k$ is not p.s.d. and maybe not even symmetric, we use the approach described for Cases 1 and/or 2.

In the present section, it has been shown how the algorithms for maximizing $\text{H}_1(X)$, $\text{h}_1(\mathbf{x})$ and $\tilde{\text{h}}_1(X)$ can be modified to handle cases where A_k is not p.s.d., and thus we have completed the description of the algorithms for maximizing $\text{H}_1(X)$, $\text{h}_1(\mathbf{x})$ and $\tilde{\text{h}}_1(X)$. In the next section, we will propose algorithms for maximizing $\text{H}_2(X)$, $\text{h}_2(\mathbf{x})$ and $\tilde{\text{h}}_2(X)$. Here, no assumptions will be made with respect to the matrices A_k at all.

Maximizing $\text{H}_2(X)$, $\text{h}_2(\mathbf{x})$ and $\tilde{\text{h}}_2(X)$

Maximizing $H_2(X)$

As has been mentioned above, we consider maximization of the function $\text{H}_2(X) = \sum_k \text{tr}\,(X'A_kX)(X'A'_kX)(X'C_kX)^{-1}$ subject to $X'X = I$. Without constraints, the function has no maximum. Obviously, $X'X = I$ is not the only possible constraint. An alternative, and at first sight meaningful constraint is $\text{tr}\,X'X = 1$. However, it can be proven that, subject to this constraint, the function has no maximum either, but only a supremum. Still other constraints are conceivable, but we limit ourselves to the constraint $X'X = I$.

As has been mentioned above, to decrease $-\mathrm{H}_2(X)$ we can decrease $\mathrm{G}_2(X)$, see (13). The function $\mathrm{G}_2(X)$ can be written as

$$\mathrm{G}_2(X) = -2 \sum_{k=1}^{p} \operatorname{tr} A_k X W_k X' + \sum_{k=1}^{p} \operatorname{tr} C_k X W_k' W_k X', \tag{25}$$

where $W_k = X_0' A_k' X_0 (X_0' C_k X_0)^{-1}$. To minimize $\mathrm{G}_2(X)$ subject to $X'X = I$, a closed-form solution does not seem available. However, it is readily seen that $\mathrm{G}_2(X)$ is a special case of the general function considered by Kiers and ten Berge (1992). Hence, we can use their procedure to decrease $\mathrm{G}_2(X)$. Specifically, according to Kiers and ten Berge (pp. 374–375), we first have to compute the matrix

$$M = 2 \sum_{k=1}^{p} \lambda_k X_0 - \sum_{k=1}^{p} (-2A_k) X_0 W_k - \sum_{k=1}^{p} (-2A_k)' X_0 W_k' + 2 \sum_{k=1}^{p} \rho_k X_0 W_k' W_k - 2 \sum_{k=1}^{p} C_k X_0 W_k' W_k, \tag{26}$$

where λ_k is taken larger than or equal to the first eigenvalue of the symmetric part of $(-2A_k) \otimes (W_k')$ and ρ_k is taken larger than or equal to the first eigenvalue of C_k, $k = 1, \ldots, p$. Next, we compute the SVD $M = PDQ'$ and update X as $X = PQ'$.

We have now described a procedure for monotonically increasing $\mathrm{H}_2(X)$ subject to $X'X = I$ (as summarized in Appendix A). If $\mathrm{H}_2(X)$ is bounded above, this procedure converges monotonically to a stable function value. It can be proven as follows that, subject to $X'X = I$, $\mathrm{H}_2(X)$ is indeed bounded above. Defining $Y_k \equiv C_k^{1/2} X (X' C_k X)^{-1/2}$, which is columnwise orthonormal, and using the SVD $C_k^{-1/2} A_k \equiv U \Delta V'$, we have, for each term of H_2, $\operatorname{tr} (X'C_kX)^{-1}(X'A_kX)(X'A_k'X) = \operatorname{tr} Y_k' C_k^{-1/2} A_k X X' A_k' C_k^{-1/2} Y_k = \operatorname{tr} Y_k' U \Delta V' X X' V \Delta U' Y_k$. According to a result by ten Berge (1983), this trace is smaller than or equal to the sum of the first r diagonal elements of the fixed matrix Δ^2, from which it follows that $\mathrm{H}_2(X)$ has a fixed upper bound. Hence, the above algorithm must converge to a stable function value. An alternative way of proving that $\mathrm{H}_2(X)$ is bounded is by noting that $\mathrm{H}_2(X)$ is a continuous function on a compact set.

An Algorithm for Maximizing $\mathrm{h}_2(\mathbf{x})$*: The Special Case Where X is a Vector*

In the special case where $X = \mathbf{x}$, the function H_2 reduces to $\mathrm{h}_2(\mathbf{x}) = \sum_k \operatorname{tr} (\mathbf{x}' A_k \mathbf{x})^2 (\mathbf{x}' C_k \mathbf{x})^{-1}$, and the update for $\mathbf{x}$ (subject to $\mathbf{x}'\mathbf{x} = 1$) can be obtained by applying the above algorithm for maximizing $\mathrm{H}_2(X)$. However, an alternative procedure, employing a closed-form solution for *minimizing* (rather than merely decreasing) G_2 is possible now. Specifically, in this case, W_k reduces to the scalar $w_k = (\mathbf{x}_0' C_k \mathbf{x}_0)^{-1} (\mathbf{x}_0' A_k \mathbf{x}_0)$, and we have to minimize the majorizing function

$$\mathrm{g}_2(\mathbf{x}) = -2 \sum_{k=1}^{p} \mathbf{x}' w_k A_k \mathbf{x} + \sum_{k=1}^{p} \mathbf{x}' w_k^2 C_k \mathbf{x}, \tag{27}$$

subject to $\mathbf{x}'\mathbf{x} = 1$. The minimum of $\mathrm{g}_2(\mathbf{x})$ is obtained by taking $\mathbf{x}$ as the first eigenvector of the symmetric part of $\sum_k (2w_k A_k - w_k^2 C_k)$. Because the function $\mathrm{h}_2(\mathbf{x})$, constrained by $\mathbf{x}'\mathbf{x} = 1$, is bounded above, we obtain a monotonically convergent algorithm by iteratively updating $\mathbf{x}$ in this way. The algorithm is summarized in Appendix A.

An Algorithm for Maximizing $\tilde{\mathrm{h}}_2(X)$

The maximization of $\tilde{\mathrm{h}}_2(X) = \sum_l \mathrm{h}_2(\mathbf{x}_l) = \sum_k \sum_l (\mathbf{x}_l'A_k\mathbf{x}_l)^2(\mathbf{x}_l'C_k\mathbf{x}_l)^{-1}$ can be handled by minimizing a function that majorizes $-\tilde{\mathrm{h}}_2(X)$. Such a function is given by

$$\tilde{\mathrm{g}}_2(X) \equiv \sum_{l=1}^{r} \mathrm{g}_2(\mathbf{x}_l) = -2 \sum_{k=1}^{p} \sum_{l=1}^{r} \mathbf{x}_l' w_{kl} A_k \mathbf{x}_l + \sum_{k=1}^{p} \sum_{l=1}^{r} \mathbf{x}_l' w_{kl}^2 C_k \mathbf{x}_l, \tag{28}$$

see (27), where $w_{kl} = (\mathbf{x}_{l0}'C_k\mathbf{x}_{l0})^{-1}(\mathbf{x}_{l0}'A_k\mathbf{x}_{l0})$. This function can be rewritten as

$$\tilde{\mathrm{g}}_2(X) = -2 \sum_{k=1}^{p} \operatorname{tr} X'A_kXW_k + \sum_{k=1}^{p} \operatorname{tr} X'C_kXW_k^2, \tag{29}$$

where W_k denotes the diagonal matrix with diagonal elements w_{kl}. This function is a special case of the one examined by Kiers and ten Berge (1992), and can be decreased using their procedure. Specifically, we take

$$M = 2 \sum_{k=1}^{p} \lambda_k X_0 + 2 \sum_{k=1}^{p} (A_k + A_k')X_0 W_k + 2 \sum_{k=1}^{p} \rho_k X_0 W_k^2 - 2 \sum_{k=1}^{p} C_k X_0 W_k^2, \tag{30}$$

where λ_k is taken larger than or equal to the first eigenvalue of the symmetric part of $(-2A_k) \otimes (W_k)$, and ρ_k is taken larger than or equal to the first eigenvalue of C_k, $k = 1, \ldots, p$. Next, we use the SVD $M = PDQ'$ and update X by taking $X = PQ'$. Again, the boundedness of the function guarantees that the procedure of iteratively updating X in the above described way converges to a stable function value.

In the special case where $C_k = I$, $k = 1, \ldots, p$, we have $\tilde{\mathrm{h}}_2(X) = \sum_k \sum_l (\mathbf{x}_l'A_k\mathbf{x}_l)^2$, which is the function maximized by INDSCAL subject to columnwise orthonormality of X (e.g., see, Kiers, 1991, p. 203). In that case, $M = 2\lambda_k X_0 + 2 \sum_k (A_k + A_k')XW_k$. If, moreover, A_k is p.s.d., we can take $\lambda_k = 0$, $k = 1, \ldots, p$, obtain M as $4 \sum_k A_kXW_k$, and obtain X as PQ' from the SVD $M = PDQ'$. This algorithm is equivalent to the one derived by ten Berge, Knol and Kiers (1988).

Convergence Properties and Performance of the Algorithms

In the descriptions of the algorithms given above, it has consistently been mentioned that the algorithms converge monotonically to a stable function value. This, by itself, does not imply that the parameter matrices converge to an accumulation point, nor even that $(X^{i+1} - X^i)$ tends to zero as $i \to \infty$. In Appendix B, it is proven that, under mild assumptions, the *differences* between update and predecessor indeed converge to zero. This implies that X converges to a single limit point or a continuum of limit points (Ostrowski, 1969, p. 203). By straightforward but tedious matrix algebraic manipulations, it can be verified that these limit points satisfy the first order necessary conditions (stationary equations) for the maximization problems at hand. As a result, all limit points are local maxima, local minima, or saddle points. Because our algorithms *increase* the function value, they can hit local *minima* only if they happen to land in a matrix X that satisfies the stationary equations exactly, which can be excluded for all practical purposes. The algorithm may converge to a saddle point, but this also seems unlikely to happen in practice, since small perturbations may be sufficient to make the algorithm find a function value above the saddle point and hence cause the algorithm to continue, after having slowed down near the saddle point. For these reasons, the algorithms will in most cases converge to local maxima. Whether these are also global maxima has to be found out by using several restarts. In the small simulation study reported below, it is studied how severe the local minimum problem is for each of the algorithms.

The algorithms proposed in the present paper have all been programmed in PCMATLAB, according to their description in Appendix A. The algorithms were tested on a PC with an 80486 processor, by applying them to 40 data sets. Specifically, matrices $A_1, \ldots, A_p$ and $C_1, \ldots, C_p$ were constructed using $p = 2$ or 4 and $n = 10$ or 20, with five replications in each condition. The matrices A_k were random p.s.d. matrices (constructed as the inner product moments of random $n \times n$ matrices) and the matrices C_k were random p.s.d. matrices to which the matrix I_n was added to prevent near singularity. The data were analyzed (using $r = 3$ or 6) by all algorithms using one "rational start," which was based on the maximum of tr $(X' \sum_k A_k X)(X' \sum_k C_k X)^{-1}$, and four random starts. Each run was considered converged if consecutive function values differed by less than .0001%. In Table 2 the average computation times and numbers of iterations (in all conditions) for one run of an algorithm are reported. In addition, it was studied how often the algorithms hit local maxima. Although we do not know whether or not a function value is the global maximum, a simple comparison of function values does indicate whether a function value is likely to be a local maximum. We considered a value a local maximum if it differed more than .1% from the highest function value for this data set. The incidence rates of such local maxima are reported in Table 2 as well (where each cell pertains to 25 runs of the algorithm, of which at most 20 can lead to what we called a local maximum). It can be seen that the incidence rates range from 0 to 10. The average numbers of iterations are between 17 and 493. The computation times, ranging from 3 to 92 seconds, depend primarily on the sizes of the arrays. The 'rational' start did not systematically outperform the other starts.

Application to Some Practical Optimization Problems

Application 1: Reciprocal PCA

Nierop's (1993, p. 95) "Reciprocal PCA" problem consists of maximizing

$$f_1(\mathbf{x}) = \left(\sum_{k=1}^{p} \frac{\mathbf{x}'\mathbf{x}}{\mathbf{x}'S_k\mathbf{x}} \right)^{-1}, \tag{31}$$

over $\mathbf{x}$, where S_k denotes a p.s.d. matrix. We will assume that S_k is p.d. here and solve the problem as follows. This problem is equivalent to minimizing $\sum_k (\mathbf{x}'\mathbf{x})(\mathbf{x}'S_k\mathbf{x})^{-1}$, which is a special case of $h_1(\mathbf{x})$. This minimization, in turn, is equivalent to maximizing

$$f_1(\mathbf{x}) = \sum_{k=1}^{p} \frac{\mathbf{x}'\tilde{A}_k\mathbf{x}}{\mathbf{x}'S_k\mathbf{x}}, \tag{32}$$

where $\tilde{A}_k = (-I - \alpha_k S_k)$ and α_k is taken smaller than or equal to the smallest eigenvalue of $S_k^{-1/2}(-I)S_k^{-1/2} = -S_k^{-1}$, thus rendering $\tilde{A}_k$ p.s.d. This eigenvalue is equal to the negative of the largest eigenvalue of S_k^{-1}. The largest eigenvalue of S_k^{-1}, in turn, is equal to the reciprocal of the smallest eigenvalue of S_k. Clearly, f_1 is a special

TABLE 2

Performance of the Algorithms

Incidence Rates of Local Maxima (out of 20)

n	p	r	$\tilde{h}_1$	H_1	$\tilde{h}_2$	H_2
10	2	3	1	1	3	3
10	2	6	3	0	0	2
10	4	3	9	9	5	5
10	4	6	5	2	3	0
20	2	3	4	6	10	5
20	2	6	9	0	4	3
20	4	3	3	9	3	5
20	4	6	6	0	10	6

Average Numbers of Iterations

n	p	r	$\tilde{h}_1$	H_1	$\tilde{h}_2$	H_2
10	2	3	130.0	30.4	138.7	46.2
10	2	6	308.1	17.6	147.0	27.9
10	4	3	159.1	43.4	108.0	58.4
10	4	6	288.7	44.7	177.6	72.4
20	2	3	351.9	56.1	407.7	120.4
20	2	6	492.5	44.6	390.3	112.5
20	4	3	274.3	69.7	188.4	122.2
20	4	6	406.2	60.9	332.5	146.9

Average Computation Times (in Seconds)

n	p	r	$\tilde{h}_1$	H_1	$\tilde{h}_2$	H_2
10	2	3	5.1	3.7	8.6	3.3
10	2	6	18.8	5.7	12.4	3.1
10	4	3	12.5	10.5	15.2	8.1
10	4	6	32.1	24.7	25.7	13.3
20	2	3	24.4	14.8	43.5	13.6
20	2	6	49.1	45.9	46.5	16.5
20	4	3	35.3	31.2	36.8	28.5
20	4	6	71.4	91.5	72.2	38.7

case of h_1, so we can use the algorithm for maximizing h_1 after replacing A_k by $(-I - \alpha_k S_k)$, and C_k by S_k, $k = 1, \ldots, p$.

Application 2: Binormamin

Harman (1976, p. 312) discussed Kaiser and Dickman's (1959) binormamin method which is a method for oblique simple structure rotation. It was proposed as "an attempt to correct for the "too oblique" bias of the quartimin criterion and the "too orthogonal" bias of the covarimin criterion" (Harman, p. 312). Harman has not given an algorithm for this procedure. It is possible to construct an algorithm rather easily, as follows. The binormamin method consists of minimizing the criterion

$$f_2(T) = \sum_{p=1}^{r} \sum_{q \neq p}^{r} \left[\sum_{j=1}^{n} \left(\frac{v_{jp}^2}{h_j^2}\right)\left(\frac{v_{jq}^2}{h_j^2}\right)\right]\left[\sum_{j=1}^{n} \left(\frac{v_{jp}^2}{h_j^2}\right) \sum_{j=1}^{n} \left(\frac{v_{jq}^2}{h_j^2}\right)\right]^{-1}, \tag{33}$$

where v_{jp} denotes element (j, p) of the obliquely rotated reference structure matrix, and h_j denotes the communality of variable j. If the unrotated reference structure is denoted by S, and, after division by h_j, its j-th row is denoted by $\mathbf{a}_j$, then $v_{jp}/h_j = \mathbf{a}_j'\mathbf{t}_p$, and we can write f_2 as

$$f_2(T) = \sum_{p=1}^{r} \sum_{q \neq p}^{r} \left[\sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_p)^2(\mathbf{a}_j'\mathbf{t}_q)^2\right]\left[\sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_p)^2 \sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_q)^2\right]^{-1}, \tag{34}$$

which has to be minimized over the columns of T, subject to $\mathbf{t}_p'\mathbf{t}_p = 1$, $p = 1, \ldots, r$. The problem can be handled by updating one column of T, say $\mathbf{t}_p$, at a time. For column $\mathbf{t}_p$, the problem is to minimize

$$\begin{aligned} f_2(\mathbf{t}_p) &= \sum_{q \neq p}^{r} \left[\sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_p)^2(\mathbf{a}_j'\mathbf{t}_q)^2\right]\left[\sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_p)^2 \sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_q)^2\right]^{-1} \\ &= \sum_{q \neq p}^{r} \left(\mathbf{t}_p' \sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_q)^2(\mathbf{a}_j\mathbf{a}_j')\mathbf{t}_p\right)\left(\mathbf{t}_p'\left(\sum_{j=1}^{n} (\mathbf{a}_j'\mathbf{t}_q)\right)\left(\sum_{j=1}^{n} \mathbf{a}_j\mathbf{a}_j'\right)\mathbf{t}_p\right)^{-1} \\ &= \sum_{q \neq p}^{r} (\mathbf{t}_p'A_q\mathbf{t}_p)(\mathbf{t}_p'C_q\mathbf{t}_p)^{-1}, \end{aligned} \tag{35}$$

where A_q and C_q are defined implicitly by (35). Clearly, (35) is a special case of h_1, and can be minimized over $\mathbf{t}_p$ by first replacing A_q by $\tilde{A}_q = (-A_q - \alpha_q C_q)$ and then using the update described in (16). Alternately updating all columns of T in this manner, we have an algorithm for monotonically decreasing $f_2(T)$.

Application 3: Generalized Discriminant Analysis

A straightforward generalization of discriminant analysis to the situation with more than one group of individuals, with scores on the same variables, is as follows. Let B_k denote the between groups covariance matrix and W_k the within groups covariance matrix for the k-th group. If it is desired to find a discriminant function that is common over groups (in that it employs the same discriminant weights), we have to maximize

$$\mathrm{f}_3(\mathbf{x}) = \sum_{k=1}^{p} \frac{\mathbf{x}'B_k\mathbf{x}}{\mathbf{x}'W_k\mathbf{x}}, \tag{36}$$

which is a function of the form $\mathrm{h}_1(\mathbf{x})$. Since the interpretation of a discriminant function is based on the discriminant weights, the above problem seems a promising method for simultaneous discriminant analysis in more than one group of individuals. This method will be particularly interesting in cases where it is assumed that the same discriminant functions ought to rule in the different groups. If the maximum of (36) is (almost) equal to the sum of function values obtained in separate discriminant analyses, it can be concluded that the different groups indeed support the same discriminant functions.

If more than one discriminant function is to be found, we may consider maximizing

$$\tilde{\mathrm{f}}_3(X) = \sum_{k=1}^{p} \sum_{l=1}^{r} \frac{\mathbf{x}_l'B_k\mathbf{x}_l}{\mathbf{x}_l'W_k\mathbf{x}_l},$$

subject to $X'X = I$. Unfortunately, this is no longer a straightforward generalization of discriminant analysis, because in discriminant analysis the discriminant components, rather than the discriminant weights, are taken columnwise orthonormal. This is not to say, of course, that the resulting method is meaningless. Future research is needed to study the properties of these generalizations of discriminant analysis.

Application 4: Variants of SCA

Some special cases of H_1 have appeared in the literature. Specifically, Millsap and Meredith's (1988) simultaneous components analysis (SCA) consists of maximizing $\sum_k \operatorname{tr} X'C_k^2X(X'C_kX)^{-1}$, where C_k denotes a covariance matrix. Kiers and ten Berge (1989) have provided an alternating least squares algorithm for their problem. The function at hand is the special case of H_1 where $A_k = C_k^2$; the Kiers and ten Berge algorithm is the corresponding special case of our present algorithm. Millsap and Meredith considered some variants of the SCA problem as well, for instance, that of maximizing $\sum_k \operatorname{tr} X'C_kG_kC_kX(X'C_kX)^{-1}$ (p. 126), where G_k denotes a p.s.d. weight matrix. Obviously, this function also is a special case of H_1, and hence our procedure gives a monotonically convergent algorithm for maximizing this function. A third special case is Kiers and ten Berge's (1994) variant of SCA, called SCA-S, which consists of maximizing $\sum_k \operatorname{tr} X'X(X'C_k^{-1}X)^{-1}$. The algorithm they proposed is a special case of the one proposed here for maximizing H_1.

Application 5: Simple Structure Rotation for SCA-S

The aim of Kiers and ten Berge's (1994) SCA-S method is to obtain components in p different data sets (with the same variables) such that the resulting structure matrices (S_k) are proportional to each other, and hence proportional to a single matrix, denoted as S. Their solution is determined only up to an oblique rotation. To exploit this rotational freedom, they proposed to rotate S such that the resulting matrices S_k have optimal simple structure, and developed an iterative algorithm for obtaining this rotation. One step of their algorithm amounts to maximizing

$$\mathrm{f}_4(T) = \sum_{k=l}^{p} \sum_{l=1}^{r} \frac{\mathbf{t}_l'A_l\mathbf{t}_l}{\mathbf{t}_l'B_k\mathbf{t}_l}, \tag{37}$$

where $A_l = S'D_lS$, D_l is a binary diagonal matrix, $B_k = S'R_k^{-1}S$, and R_k denotes the correlation matrix for set k. Each column of T is constrained such that $\mathbf{t}_l'\mathbf{t}_l = 1$, hence, the problem of maximizing f_4 can be separated into r separate problems of maximizing

$$f_5(\mathbf{t}_l) = \sum_{k=1}^{p} \frac{\mathbf{t}_l' A_l \mathbf{t}_l}{\mathbf{t}_l' B_k \mathbf{t}_l}, \tag{38}$$

for $l = 1, \ldots, r$. Clearly, the function f_5 is a special case of h_1. In fact, Kiers and ten Berge proposed to maximize (38) by a special case of the presently proposed algorithm for maximizing h_1.

Application 6: Set Components Analysis

Nierop (1993, p. 41) has discussed a problem which is a special case of that of maximizing $\tilde{h}_2(X)$. His problem amounts to maximizing

$$f_6(X) = \sum_{k=1}^{p} \sum_{l=1}^{r} \frac{(\mathbf{x}_l' S_k \mathbf{x}_l)^2}{\mathbf{x}_l' S_k^2 \mathbf{x}_l}, \tag{39}$$

where S_k denotes a matrix $H_k'H_k$ and H_k contains scores of observation units on the same set of variables, measured in the k-th sample. This function is the special case of $\tilde{h}_2$ with $A_k = S_k$ and $C_k = S_k^2$. It can be verified that our algorithm for maximizing $\tilde{h}_2$ is closely related, but slightly different from the procedure for updating X used by Nierop (pp. 151 ff).

Discussion

In the present paper, we have considered only the functions H_1 and H_2 (and their variants) as special cases of the general function $-H(X)$. Other special cases where $F_k(X) = (X'A_kX)^2$ and or where $s = 2$ can also be handled by decreasing $G(X)$. Suggestions on how $G(X)$ can be decreased in those cases can be obtained from the author.

The performance of the present algorithms seems quite satisfactory. There is a mild local maximum problem, but when a large number of restarts is taken, the chance of hitting a local optimum is small. The 'rational' start employed in the simulation study did not outperform the random starts. However, in specific contexts it may be possible to obtain other, more rational starts than the one considered here, and these may further enhance the chance of finding the global maximum. In general, the algorithms require considerable, but not prohibitive numbers of iterations. As with other majorization based algorithms, it can be expected that the algorithms converge linearly, with a relatively slow convergence rate. This was confirmed by the iterative processes for the first runs of the algorithms when applied to the data sets of our simulation study.

We have hinted at rotational or even transformational indeterminacy in the solutions for the H_1 and H_2 maximization problems. It can easily be seen that X in H_1 can be determined only up to a nonsingular transformation, and that the X in H_2 can be determined only up to an orthonormal rotation, because such transformations cancel in the function formulas. For the X that yields the maximum of $\tilde{h}_1$ or $\tilde{h}_2$ it seems that such transformations are generally not allowed. In fact, in the analyses of the 40 data sets from the simulation studies, considering the first three starts only, we found that whenever two equal function values were found (as happened in 37 cases for both algorithms), the corresponding solutions were equal as well (up to permutation). Unfortunately, attempts to prove this type of uniqueness have failed so far.

Dinkelbach (1967) has described a procedure for maximizing a quotient of two functions which is, at first glance, similar to certain of the problems discussed in the present paper. Specifically, he has shown that one can monotonically improve $f(\mathbf{x})/g(\mathbf{x})$, where f and g are, for instance, quadratic forms, by taking $\mathbf{x}$ such that it maximizes $f(\mathbf{x}) - f(\mathbf{x}_0)(g(\mathbf{x}_0))^{-1}g(\mathbf{x})$. However, his method cannot be generalized straightforwardly to that of maximizing a sum of quotients of such functions. Moreover, even in the case of maximizing only one quotient his procedure differs from ours. Specifically, if $f(\mathbf{x}) = \mathbf{x}'A\mathbf{x}$ and $g(\mathbf{x}) = \mathbf{x}'C\mathbf{x}$, his algorithm boils down to taking the first eigenvector of $A - (\mathbf{x}_0'A\mathbf{x}_0)(\mathbf{x}_0'C\mathbf{x}_0)^{-1}C$. Our algorithm for maximizing h_1 for $p = 1$ updates $\mathbf{x}$ according to (15) as $\mathbf{x} = (\mathbf{x}_0'C\mathbf{x}_0)(\mathbf{x}_0'A\mathbf{x}_0)^{-1}C^{-1}A\mathbf{x}_0$, possibly normalized to a unit sum of squares. This method comes down to the well-known power method applied to $C^{-1}A$. Clearly, Dinkelbach's method is much more involved since it requires the full computation of a first eigenvector in each iteration.

Appendix A

A Summary of the Algorithms for Maximizing the Six Functions of Table 1

Maximization of H_1

1. $A_k := \frac{1}{2}(A_k + A_k'),\ k = 1, \ldots, p$
2. $\alpha_k := \max(0, \mu_1(C_k^{-1/2}A_kC_k^{-1/2}))$, where $\mu_1(\cdot)$ denotes "smallest eigenvalue of," $k = 1, \ldots, p$
3. $A_k := A_k - \alpha_k C_k$
4. Initialize X (e.g., randomly) and ε (e.g., 0.000001)
5. $\mathrm{H}_1 := \sum_{k=1}^{p} \mathrm{tr}\,(X'A_kX)(X'C_kX)^{-1}$

Iterative part

6. $X_0 := X$; $\mathrm{H}_1^{\mathrm{old}} := \mathrm{H}_1$
7. $W_k'W_k := (X_0'C_kX_0)^{-1}(X_0'A_kX_0)(X_0'C_kX_0)^{-1},\ k = 1, \ldots, p$
8. $\mathrm{Vec}\,(X) := (\sum_{k=1}^{p}(W_k'W_k)\otimes C_k)^{-1}\sum_{k=1}^{p}\mathrm{Vec}(A_kX_0(X_0'C_kX_0)^{-1})$
9. $X := \mathrm{Mat}\,(X)$, where Mat $(\cdot)$ denotes the inverse operation of Vec $(\cdot)$
10. (optional:) orthonormalize X by the Gram-Schmidt procedure
11. $\mathrm{H}_1 := \sum_{k=1}^{p} \mathrm{tr}\,(X'A_kX)(X'C_kX)^{-1}$
12. If $\mathrm{H}_1 - \mathrm{H}_1^{\mathrm{old}} > \mathrm{H}_1\varepsilon$ go to Step 6, else consider the algorithm converged
13. After convergence, compute the value of the original function as $\mathrm{H}_1 + r\sum_k \alpha_k$.

Maximization of h_1

Steps 1 through 3 as in the maximization of H_1

4. Initialize $\mathbf{x}$ (e.g., randomly) and ε (e.g., 0.000001)
5. $\mathrm{h}_1 := \sum_{k=1}^{p}(\mathbf{x}'A_k\mathbf{x})(\mathbf{x}'C_k\mathbf{x})^{-1}$

Iterative part

6. $\mathbf{x}_0 := \mathbf{x}$; $\mathrm{h}_1^{\mathrm{old}} := \mathrm{h}_1$
7. $\mathbf{x} := (\sum_{k=1}^{p}(\mathbf{x}_0'C_k\mathbf{x}_0)^{-2}(\mathbf{x}_0'A_k\mathbf{x}_0)C_k)^{-1}\sum_{k=1}^{p}(\mathbf{x}_0'C_k\mathbf{x}_0)^{-1}A_k\mathbf{x}_0$
8. (optional:) normalize $\mathbf{x}$ to unit length: $\mathbf{x} := \mathbf{x}/(\mathbf{x}'\mathbf{x})^{1/2}$
9. $\mathrm{h}_1 := \sum_{k=1}^{p}(\mathbf{x}'A_k\mathbf{x})(\mathbf{x}'C_k\mathbf{x})^{-1}$

10. If $h_1 - h_1^{old} > h_1 \varepsilon$ go to Step 6, else consider the algorithm converged
11. After convergence, compute the value of the original function as $h_1 + \sum_k \alpha_k$.

Maximization of $\bar{h}_1$

Steps 1 through 3 as in the maximization of H_1.

4. $\rho_k := \lambda_1(C_k)$, where $\lambda_1(\cdot)$ denotes "first eigenvalue of," $k = 1, \ldots, p$
5. Initialize X (e.g., randomly, such that $X'X = I$) and ε (e.g., 0.000001)
6. $\bar{h}_1 := \sum_{k=1}^{p} \text{tr}\,(\text{Diag}\,(X'A_kX))(\text{Diag}\,(X'C_kX))^{-1}$

Iterative part

7. $X_0 := X$; $\bar{h}_1^{old} := \bar{h}_1$
8. $D_k := (\text{Diag}\,(X_0'C_kX_0))^{-2}\,\text{Diag}\,(X_0'A_kX_0)$, $k = 1, \ldots, p$
9. $M := \sum_{k=1}^{p} \rho_k X_0 D_k - \underline{\sum_{k=1}^{p} C_k X_0 D_k}\; \sum_{k=1}^{p} A_k X_0(\text{Diag}(X_0'C_kX_0))^{-1}$
10. SVD of M: $M = PDQ'$
11. $X := PQ'$
12. $\bar{h}_1 := \sum_{k=1}^{p} \text{tr}\,(\text{Diag}\,(X'A_kX))(\text{Diag}\,(X'C_kX))^{-1}$
13. If $\bar{h}_1 - \bar{h}_1^{old} > \bar{h}_1 \varepsilon$ go to Step 7, else consider the algorithm converged
14. After convergence, compute the value of the original function as $\bar{h}_1 + r\sum_k \alpha_k$.

Maximization of H_2

1. Initialize X (e.g., randomly, such that $X'X = I$) and ε (e.g., 0.000001)
2. $H_2 := \sum_{k=1}^{p} \text{tr}\,(X'A_kX)(X'A_k'X)(X'C_kX)^{-1}$
3. $\rho_k := \lambda_1(C_k)$, $k = 1, \ldots, p$.

Iterative part

4. $X_0 := X$; $H_2^{old} := H_2$
5. $W_k := X_0'A_k'X_0(X_0'C_kX_0)^{-1}$, $k = 1, \ldots, p$
6. $\lambda_k := \lambda_1((-A_k) \otimes W_k' + (-A_k') \otimes W_k)$, $k = 1, \ldots, p$
7. $M := \sum_{k=1}^{p} \lambda_k X_0 + \sum_{k=1}^{p} A_k X_0 W_k + \sum_{k=1}^{p} A_k' X_0 W_k' + \sum_{k=1}^{p} \rho_k X_0 W_k' W_k$ $- \sum_{k=1}^{p} C_k X_0 W_k' W_k$
8. SVD of M: $M = PDQ'$
9. $X := PQ'$
10. $H_2 := \sum_{k=1}^{p} \text{tr}\,(X'A_kX)(X'A_k'X)(X'C_kX)^{-1}$
11. If $H_2 - H_2^{old} > H_2 \varepsilon$ go to Step 4, else consider the algorithm converged.

Maximization of h_2

1. Initialize $\mathbf{x}$ (e.g., randomly, such that $\mathbf{x}'\mathbf{x} = 1$) and ε (e.g., 0.000001)
2. $h_2 := \sum_{k=1}^{p} (\mathbf{x}'A_k\mathbf{x})^2(\mathbf{x}'C_k\mathbf{x})^{-1}$

Iterative part

3. $\mathbf{x}_0 := \mathbf{x}$; $h_2^{old} := h_2$
4. $w_k := \mathbf{x}_0'A_k\mathbf{x}_0(\mathbf{x}_0'C_k\mathbf{x}_0)^{-1}$, $k = 1, \ldots, p$
5. $M := \sum_{k=1}^{p} (w_k(A_k + A_k') - w_k^2 C_k)$
6. $\mathbf{x}$:= first eigenvector of M
7. $h_2 := \sum_{k=1}^{p} (\mathbf{x}'A_k\mathbf{x})^2(\mathbf{x}'C_k\mathbf{x})^{-1}$
8. If $h_2 - h_2^{old} > h_2 \varepsilon$ go to Step 3, else consider the algorithm converged.

Maximization of $\tilde{h}_2$

1. Initialize X (e.g., randomly, such that $X'X = I$) and ε (e.g., 0.000001)
2. $\rho_k := \lambda_1(C_k)$, $k = 1, \ldots, p$.
3. $\tilde{h}_2 := \sum_{k=1}^{p} \text{tr}\,(\text{Diag}\,(X'A_kX))^2(\text{Diag}\,(X'C_kX))^{-1}$

Iterative part

4. $X_0 := X$; $\tilde{h}_2^{\text{old}} := \tilde{h}_2$
5. $W_k := (\text{Diag}\,(X_0'A_kX_0))(\text{Diag}\,(X_0'C_kX_0))^{-1}$, $k = 1, \ldots, p$
6. $\lambda_k := \lambda_1\,((-A_k) \otimes W_k + (-A_k') \otimes W_k')$, $k = 1, \ldots, p$
7. $M := \sum_{k=1}^{p} \lambda_kX_0 + \sum_{k=1}^{p} (A_k + A_k')X_0W_k + \sum_{k=1}^{p} \rho_kX_0W_k^2 - \sum_{k=1}^{p} C_kX_0W_k^2$
8. $X := PQ'$
9. $\tilde{h}_2 := \sum_{k=1}^{p} \text{tr}\,(\text{Diag}\,(X'A_kX))^2(\text{Diag}\,(X'C_kX))^{-1}$
10. If $\tilde{h}_2 - \tilde{h}_2^{\text{old}} > \tilde{h}_2\varepsilon$ go to Step 4, else consider the algorithm converged.

Appendix B

Convergence Properties of the Algorithms for Maximizing H_1, h_1, $\tilde{h}_1$, H_2, h_2 and $\tilde{h}_2$

In this appendix, it will be proven that, under mild assumptions, all algorithms proposed in the present paper have the property that the difference between update and predecessor converges to zero. As has been mentioned, this type of convergence does not yet imply that the vectors $\mathbf{x}$ or the matrices X converge to a stable solution, but, from a result by Ostrowski (1969, p. 203) it does follow that X converges to either a stable point, or to a continuum of limit points. Also, on the basis of the convergence of differences, it can be proven that at convergence the normal equations are satisfied.

Before proving that the differences between update and predecessor converge to zero, we introduce a simplified notation, as follows. First of all, we treat all maximization problems as minimizing the negatives of the functions, and denote the latter as $H(X)$, as was already introduced in (8). In fact, $H(X)$ of (8) has $-H_1(X)$, $-h_1(\mathbf{x})$, $-H_2(X)$ and $-h_2(\mathbf{x})$ as special cases, but in the present appendix, we use $H(X)$ also to cover $-\tilde{h}_1(\mathbf{x})$ and $-\tilde{h}_2(\mathbf{x})$. To further simplify our notation, we write $h^i \equiv H(X^i)$, where X^i denotes the values of X after iteration i. Hence, the monotone convergence of the algorithms implies that $h^{i+1} \le h^i$ and $\text{Lim}_{i\to\infty} h^i$, denoted as h^∞, is a fixed, but unknown value.

In the derivations of the algorithms, it has been seen that all algorithms involve the minimization or decrease of a majorizing function $G(X)$. In the algorithms for maximizing $H_1(X)$, $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$, the majorization functions $G_1(X)$, $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$, respectively were *minimized*. In the other algorithms, for maximizing $\tilde{h}_1(X)$, $\tilde{h}_2(X)$ and $H_2(X)$, the employed majorization functions were *decreased* according to Kiers and ten Berge's (1992) procedure, which itself, in turn, is based on minimizing a majorizing function. The functions minimized by the latter procedure, denoted here as $\tilde{k}_1(X)$, $\tilde{k}_2(X)$ and $K_2(X)$, respectively, majorize $\tilde{g}_1(X)$, $\tilde{g}_2(X)$ and $G_2(X)$, respectively, and thus indirectly majorize the negatives of the original functions. To describe these majorizing functions in a general notation, we denote the primary majorizing functions as $G(X)$, with $g^i \equiv G(X^i)$, and the secondary majorizing functions as $K(X)$, with $k^i \equiv K(X^i)$.

In terms of the above notation, it follows from the majorization inequalities employed that, for all algorithms, we have $H(X^{i+1}) \le G(X^{i+1}) \le K(X^{i+1}) \le K(X^i) = G(X^i) = H(X^i)$, hence

$$h^\infty \le \cdots \le h^{i+1} \le g^{i+1} \le k^{i+1} \le k^i = g^i = h^i, \tag{40}$$

where the k-functions are used only in three of the six problems. It follows from (40) that

$$\lim_{i\to\infty} (g^i - g^{i+1}) = 0, \tag{41}$$

and

$$\lim_{i\to\infty} (k^{i+1} - g^{i+1}) = 0. \tag{42}$$

Result (41) will be used for the algorithms involving one majorizing function only; result (42) will be used for the algorithms involving both majorizing functions. It will be shown for all algorithms that it follows from either (41) or (42) that

$$\lim_{i\to\infty} (X^{i+1} - X^i) = 0, \tag{43}$$

hence that the differences between update and predecessor converge to 0.

Proof of (43) *for the Algorithms for Maximizing* $H_1(X)$ *and* $h_1(X)$

In the algorithm for maximizing $H_1(X)$, the majorization function is $G_1(X)$, as given by (14). In order to use (41), we will elaborate $(g^i - g^{i+1})$. First, we note that $G_1(X^i)$ can be written as

$$g^i = G_1(X^i) = c + \|s - T^i \operatorname{Vec}(X^i)\|^2, \tag{44}$$

where c and s are constants, defined implicitly in (14), and

$$T^i = \begin{pmatrix} W_1^i \otimes C_1^{1/2} \\ \vdots \\ W_p^i \otimes C_p^{1/2} \end{pmatrix},$$

where $W_k^i \equiv A_k^{1/2} X^i (X^{i\prime} C_k X^i)^{-1}$. The expression for $G_1(X^{i+1})$ is relatively simple, because we can use that $\operatorname{Vec}(X^{i+1}) = (T^{i\prime} T^i)^{-1} T^{i\prime} s$, assuming that T^i has full column rank. Then we find $g^{i+1} = c + \| s - T^i (T^{i\prime} T^i)^{-1} T^{i\prime} s\|^2$, and we have

$$\begin{aligned} g^i - g^{i+1} &= \|s - T^i \operatorname{Vec}(X^i)\|^2 - \|s - T^i(T^{i\prime}T^i)^{-1}T^{i\prime}s\|^2 \\ &= \operatorname{Vec}(X^i)' T^{i\prime} T^i \operatorname{Vec}(X^i) - 2s' T^i \operatorname{Vec}(X^i) + s' T^i (T^{i\prime} T^i)^{-1} T^{i\prime} s \\ &= \|T^i (T^{i\prime} T^i)^{-1} T^{i\prime} s - T^i \operatorname{Vec}(X^i)\|^2 \\ &= \|T^i \operatorname{Vec}(X^{i+1}) - T^i \operatorname{Vec}(X^i)\|^2 \\ &= (\operatorname{Vec}(X^{i+1} - X^i))' T^{i\prime} T^i (\operatorname{Vec}(X^{i+1} - X^i)). \end{aligned} \tag{45}$$

Because $\lim_{i\to\infty} (g^i - g^{i+1}) = 0$, we have $\lim_{i\to\infty} (e^{i\prime} T^{i\prime} T^i e^i) = 0$, with $e^i \equiv \operatorname{Vec}(X^{i+1} - X^i)$. If $T^{i\prime} T^i$ is nonsingular and does not tend to a singular matrix either, it follows that $\lim_{i\to\infty} (e^{i\prime} e^i) = \lim_{i\to\infty} (\operatorname{tr} (X^{i+1} - X^i)'(X^{i+1} - X^i)) = 0$, hence $\lim_{i\to\infty} (X^{i+1} - X^i) = 0$, which has to be proven. We will now show why this is indeed the case, under mild assumptions. The term $e^{i\prime} T^{i\prime} T^i e^i$ can be elaborated as

$$\mathbf{e}^{i\prime}T^{i\prime}T^{i}\mathbf{e}^{i} = \sum_{k=1}^{p} \mathbf{e}^{i\prime}((X^{i\prime}C_kX^i)^{-1}X^{i\prime}A_kX^i(X^{i\prime}C_kX^i)^{-1} \otimes C_k)\mathbf{e}^i$$

$$= \sum_{k=1}^{p} \text{tr } (C_k^{1/2}(X^{i+1} - X^i))((X^{i\prime}C_kX^i)^{-1}X^{i\prime}A_kX^i(X^{i\prime}C_kX^i)^{-1})$$

$$\cdot ((X^{i+1} - X^i)'C_k^{1/2}). \tag{46}$$

Each term in the right-hand side of (46) consists of the sum of r quadratic forms. We will focus on these quadratic forms now. Assuming that X^i is columnwise orthonormal (as will be justified later), defining the columnwise orthonormal matrix $Y_k^i \equiv C_k^{1/2}X^i(X^{i\prime}C_kX^i)^{-1/2}$, and denoting the smallest eigenvalue of a matrix by $\lambda_{\min}(\cdot)$, we have, for every vector $\mathbf{z}$,

$$\mathbf{z}'((X^{i\prime}C_kX^i)^{-1}X^{i\prime}A_kX^i(X^{i\prime}C_kX^i)^{-1})\mathbf{z}$$

$$= \mathbf{z}'X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}A_kC_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}X^i\mathbf{z}$$

$$\geq \lambda_{\min}(C_k^{-1/2}A_kC_k^{-1/2})(\mathbf{z}'X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}X^i\mathbf{z})$$

$$\geq \lambda_{\min}(C_k^{-1/2}A_kC_k^{-1/2})\lambda_{\min}(X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}X^i)(\mathbf{z}'\mathbf{z}), \tag{47}$$

where we used twice that $\mathbf{z}'S\mathbf{z} \geq \lambda_{\min}(S)(\mathbf{z}'\mathbf{z})$, for any symmetric matrix S. Next, we can use that $\lambda_{\min}(X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}X^i) = (\sigma_{\min}(X^{i\prime}C_k^{-1/2}Y_k^i))^2$, where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of the matrix between parentheses. Because X^i and Y_k^i are columnwise orthonormal, we have $\sigma_{\min}(X^{i\prime}C_k^{-1/2}Y_k^i) \geq \sigma_{\min}(C_k^{-1/2})$, and it follows that

$$\mathbf{z}'(X^{i\prime}C_kX^i)^{-1}X^{i\prime}A_kX^i(X^{i\prime}C_kX^i)^{-1}\mathbf{z} \geq \lambda_{\min}(C_k^{-1/2}A_kC_k^{-1/2})$$

$$(\sigma_{\min}(C_k^{-1/2}))^2(\mathbf{z}'\mathbf{z}) \equiv c_k(\mathbf{z}'\mathbf{z}), \tag{48}$$

where c_k is a nonnegative constant, implicitly defined by (48). Note that, because C_k is p.d., the constant c_k is positive unless $\lambda_{\min}(A_k) = 0$. Combining (46) and (48), we have

$$\mathbf{e}^{i\prime}T^{i\prime}T^{i}\mathbf{e}^{i} \geq \sum_{k=1}^{p} c_k \text{ tr } (C_k^{1/2}(X^{i+1} - X^i))((X^{i+1} - X^i)'C_k^{1/2}), \tag{49}$$

and it follows from $\text{Lim}_{i\to\infty} (\mathbf{e}^{i\prime}T^{i\prime}T^i\mathbf{e}^i) = 0$ that $\text{Lim}_{i\to\infty} (\sum_k c_k\|C_k^{1/2}(X^{i+1} - X^i)\|^2) = 0$. As soon as one of the constants c_k is strictly positive, we have $\text{Lim}_{i\to\infty} (\|C_k^{1/2}(X^{i+1} - X^i)\|^2) = 0$. Hence, because C_k is p.d., $\text{Lim}_{i\to\infty} (X^{i+1} - X^i) = 0$. The assumption that at least one c_k is positive will usually be satisfied. This is because for all c_k to be zero, we must have $\lambda_{\min}(A_k) = 0$ for all k. It is very unlikely to have $\lambda_{\min}(A_k) = 0$ for all k, in practice, and, moreover, it can always be avoided by adding εC_k to A_k, see (23), which does not change the maximization problem (where ε is a small positive value).

It has been shown above that $\text{Lim}_{i\to\infty} (X^{i+1} - X^i) = 0$ *assuming* that X^i is columnwise orthonormal. Of course, this is a very strong assumption. However, in the description of our algorithm it has already been mentioned that we can, without affecting the function value, orthonormalize X (assuming that it has full rank) after updating it. It follows that, if X^i has come about by orthonormalizing the i-th least squares update of X ($\tilde{X}^i$ say), then $\text{Lim}_{i\to\infty}$ $(\tilde{X}^{i+1} - X^i) = 0$. We can express X^{i+1} in terms of $\tilde{X}^{i+1}$ as $X^{i+1} = \text{GS}(\tilde{X}^{i+1})$ where GS(·) denotes applying the Gram-Schmidt transformation. Obviously, as $\tilde{X}^{i+1}$ tends to X^i, $\tilde{X}^{i+1}$ tends to a columnwise orthonormal matrix, and $X^{i+1} = \text{GS}(\tilde{X}^{i+1})$ tends to $\tilde{X}^{i+1}$. Hence, from $\text{Lim}_{i\to\infty}$ $(\tilde{X}^{i+1} - X^i) = 0$ it follows that $\text{Lim}_{i\to\infty}$ $(X^{i+1} - X^i) = 0$.

For $\text{h}_1(\text{x})$, we can use the proof given above, because the algorithm is a special case of the algorithm for maximizing $\text{H}_1(X)$.

Proof of (43) *for the Algorithms for Maximizing* $\tilde{\text{h}}_1(X)$, $\text{H}_2(X)$ *and* $\tilde{\text{h}}_2(X)$

The algorithms for maximizing $\tilde{\text{h}}_1(X)$, $\text{H}_2(X)$ and $\tilde{\text{h}}_2(X)$ are based on decreasing the functions $\tilde{\text{g}}_1(X)$, $\text{G}_2(X)$ and $\tilde{\text{g}}_2(X)$, respectively, by the majorization approach given by Kiers and ten Berge (1992). Specifically, the functions are majorized by functions $\tilde{\text{k}}_1(X)$, $\text{K}_2(X)$ and $\tilde{\text{k}}_2(X)$, respectively. Without giving these functions explicitly, we will only give the expression for $\text{k}^{i+1} - \text{g}^{i+1}$, that can be derived directly from Kiers and ten Berge (pp. 372–373). Specifically, their formulas (3) and (4) give the inequalities that constitute the inequality $\text{k}(X^{i+1}) \geq \text{g}(X^{i+1})$. Hence, the difference $\text{k}(X^{i+1}) - \text{g}(X^{i+1})$ can be derived by computing the differences in right- and left hand side of inequalities (3) and (4), as will be elaborated below, for each problem separately.

For the problem of maximizing $\tilde{\text{h}}_1(X)$, we have

$$\text{k}^{i+1} - \text{g}^{i+1} = \text{k}(X^{i+1}) - \text{g}(X^{i+1}) = \sum_{k=1}^{p} \rho_k \text{ tr } E^i D_k^i E^{i\prime}$$

$$- \sum_{k=1}^{p} \text{tr } E^{i\prime} C_k E^i D_k^i = \sum_{k=1}^{p} \text{tr } D_k^{i\,1/2} E^{i\prime} (\rho_k I - C_k) E^i D_k^{i\,1/2}, \quad (50)$$

where $E^i = (X^{i+1} - X^i)$, $D_k^i = \text{Diag}\,(X^{i\prime} A_k X^i)(\text{Diag}\,(X^{i\prime} C_k X^i))^{-2}$, which is p.s.d., and ρ_k is a value larger than or equal to the first eigenvalue of C_k. From the choice of ρ_k it follows that $(\rho_k I - C_k)$ is p.s.d., or even p.d., if ρ_k is larger than the first eigenvalue of C_k. Hence each term in (50) is nonnegative. Therefore, it follows from (42) that $\text{Lim}_{i\to\infty}$ $(\text{tr } D_k^{i\,1/2} E^{i\prime} (\rho_k I - C_k) E^i D_k^{i\,1/2}) = 0$, for every k. If $(\rho_k I - C_k)$ is p.d. for a certain k (as can always be arranged by choosing ρ_k larger than the first eigenvalue), it follows at once that $\text{Lim}_{i\to\infty}$ $(\text{tr } E^i D_k^i E^{i\prime}) = 0$. If $(\rho_k I - C_k)$ is not p.d. for any k (as is the case when ρ_k is chosen *equal* to the largest eigenvalue of C_k, $k = 1, \ldots, p$), it follows from (42) that the columns of $E^i D_k^{i\,1/2}$ are orthogonal to the columns of $(\rho_k I - C_k)$, for every k. In practice, it is highly unlikely that the matrices $(\rho_k I - C_k)$ have a nonempty intersection of their null spaces. Hence, it can be assumed that, even when $(\rho_k I - C_k)$ is not p.d., $k = 1, \ldots, p$, we still have $\text{Lim}_{i\to\infty}$ $(E^i D_k^{i\,1/2}) = 0$, hence $\text{Lim}_{i\to\infty}$ $(\text{tr } E^i D_k^i E^{i\prime}) = 0$. Having proven that $\text{Lim}_{i\to\infty}$ $(\text{tr } E^i D_k^i E^{i\prime}) = 0$, it still remains to be proven that $\text{Lim}_{i\to\infty}$ $(\text{tr } E^i E^{i\prime}) = 0$. To do so, we study the matrices D_k^i. From the definition of D_k^i, it follows for its j-th diagonal element, $d_{jk}^i = (\mathbf{x}_j^{i\prime} A_k \mathbf{x}_j^i)(\mathbf{x}_j^{i\prime} C_k \mathbf{x}_j^i)^{-2}$, that

$$d_{jk}^i \geq \lambda_{\min}(A_k)(\lambda_1(C_k))^{-2} \equiv \nu_k, \quad (51)$$

hence $\text{tr } (E^i D_k^i E^{i\prime}) \geq \nu_k \text{ tr } (E^{i\prime} E^i)$. Assuming that A_k is nonsingular for at least one k, and hence that $\nu_k > 0$ for this k, it follows from $\text{Lim}_{i\to\infty}$ $(\text{tr } E^i D_k^i E^{i\prime}) = 0$ that that $\text{Lim}_{i\to\infty}$ $(\text{tr } E^{i\prime} E^i) = 0$.

For the problem of maximizing $H_2(X)$, we have from Kiers and ten Berge's (1992) formulas (3) and (4) that

$$\mathrm{k}^{i+1} - \mathrm{g}^{i+1} = \sum_{k=1}^{p} \lambda_k^i \operatorname{tr} E^{i\prime}E^i - \sum_{k=1}^{p} \operatorname{tr} E^{i\prime}(-2A_k)E^iW_k^i + \sum_{k=1}^{p} \rho_k \operatorname{tr} E^iW_k^{i\prime}W_k^iE^{i\prime}$$

$$- \sum_{k=1}^{p} \operatorname{tr} E^{i\prime}C_kE^iW_k^{i\prime}W_k^i = \mathbf{e}^{i\prime} \sum_{k=1}^{p} (\lambda_k^iI - (-A_k \otimes W_k^{i\prime} - A_k' \otimes W_k^i))\mathbf{e}^i$$

$$+ \sum_{k=1}^{p} \operatorname{tr} W_k^iE^{i\prime}(\rho_kI - C_k)E^iW_k^{i\prime}, \quad (52)$$

where $W_k^i = X^{i\prime}A_k'X^i(X^{i\prime}C_kX^i)^{-1}$, λ_k^i is taken larger than or equal to the first eigenvalue of $(-A_k \otimes W_k^{i\prime} - A_k' \otimes W_k^i)$, E^i and ρ_k are defined as before, and $\mathbf{e}^i \equiv \operatorname{Vec}(E^{i\prime})$. Because all terms in (52) are nonnegative, it follows from (42) that, for instance, $\operatorname{Lim}_{i\to\infty} (\operatorname{tr} W_k^iE^{i\prime}(\rho_kI - C_k)E^iW_k^{i\prime}) = 0$, for all k. Assuming that the matrices $(\rho_kI - C_k)$, $k = 1, \ldots, p$, have an empty intersection of null spaces, it follows that $\operatorname{Lim}_{i\to\infty} (\operatorname{tr} W_k^iE^{i\prime}E^iW_k^{i\prime}) = 0$. In order to use the latter result, we need an expression for $\mathbf{z}'W_k^{i\prime}W_k^i\mathbf{z}$ for arbitrary vectors $\mathbf{z}$. Recalling the definition of the columnwise orthonormal matrix $Y_k^i \equiv C_k^{1/2}X^i(X^{i\prime}C_kX^i)^{-1/2}$, see (47), we have for every $\mathbf{z}$

$$\begin{aligned}
\mathbf{z}'W_k^{i\prime}W_k^i\mathbf{z} &= \mathbf{z}'(X^{i\prime}C_kX^i)^{-1}(X^{i\prime}A_kX^i)(X^{i\prime}A_k'X^i)(X^{i\prime}C_kX^i)^{-1}\mathbf{z} \\
&= \mathbf{z}'X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}A_kX^iX^{i\prime}A_k'C_k^{-1/2}Y_i^iY_k^{i\prime}C_k^{-1/2}X^i\mathbf{z} \\
&\geq (\mathbf{z}'X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}X^i\mathbf{z})\lambda_{\min}(Y_k^{i\prime}C_k^{-1/2}A_kX^iX^{i\prime}A_k'C_k^{-1/2}Y_k^{i\prime}) \\
&\geq (\mathbf{z}'\mathbf{z})\lambda_{\min}(X^{i\prime}C_k^{-1/2}Y_k^iY_k^{i\prime}C_k^{-1/2}X^i)(\sigma_{\min}(Y_k^{i\prime}C_k^{-1/2}A_kX^i))^2 \\
&= (\mathbf{z}'\mathbf{z})(\sigma_{\min}(X^{i\prime}C_k^{-1/2}Y_k^i)^2(\sigma_{\min}(Y_k^{i\prime}C_k^{-1/2}A_kX^i))^2 \\
&\geq (\mathbf{z}'\mathbf{z})(\sigma_{\min}(C_k^{-1/2}))^2(\sigma_{\min}(C_k^{-1/2}A_k))^2, \qquad (53)
\end{aligned}$$

which is strictly larger than 0 for at least one k, if the corresponding A_k is nonsingular. Hence, from $\operatorname{Lim}_{i\to\infty} (\operatorname{tr} E^iW_k^{i\prime}W_k^iE^{i\prime}) = 0$, we have $\operatorname{Lim}_{i\to\infty} (\operatorname{tr} E^iE^{i\prime}) = 0$.

For the problem of maximizing $\tilde{\mathrm{h}}_2(X)$, we can derive analogously to the derivation of (52) that

$$\mathrm{k}^{i+1} - \mathrm{g}^{i+1} = \mathbf{e}^{i\prime} \sum_{k=1}^{p} (\lambda_k^iI - (-A_k \otimes W_k^i - A_k' \otimes W_k^i))\mathbf{e}^i$$

$$+ \sum_{k=1}^{p} \operatorname{tr} W_k^iE^{i\prime}(\rho_kI - C_k)E^iW_k^i, \quad (54)$$

where $W_k^i = \operatorname{Diag}(X^{i\prime}A_kX^i)(\operatorname{Diag}(X^{i\prime}C_kX^i))^{-1}$. As above, it follows from (43) that $\operatorname{Lim}_{i\to\infty} (\operatorname{tr} W_k^iE^{i\prime}E^iW_k^{i\prime}) = 0$, and similarly, assuming that at least one of the matrices A_ka is nonsingular, it follows that $\operatorname{Lim}_{i\to\infty} (\operatorname{tr} E^iE^{i\prime}) = 0$.

Proof of (43) for the Algorithm for Maximizing $h_2(x)$

The algorithm for maximizing $h_2(x)$ is based on global minimization of the function $g_2(x)$, see (27), subject to $x'x = 1$. The minimum is found by taking x equal to the last eigenvector of $\sum_k (w_k^{i2} C_k - w_k^i(A_k + A'_k))$, where $w_k^i = x^{i\prime} A_k x^i (x^{i\prime} C_k x^i)^{-1}$, $k = 1, \ldots, p$. Elaborating $(g^i - g^{i+1})$, we find

$$g^i - g^{i+1} = x^{i\prime}\left(\sum_{k=1}^{p} (w_k^i)^2 C_k - w_k^i(A_k + A'_k)\right)x^i - \lambda_{\min}\left(\sum_{k=1}^{p} ((w_k^i)^2 C_k - w_k^i(A_k + A'_k))\right). \tag{55}$$

Let $S_i \equiv \sum_k ((w_k^i)^2 C_k - w_k^i(A_k + A'_k) - \lambda_m I)$, where λ_m abbreviates the expression with $\lambda_{\min}(\cdot)$ in (55). Using that x^{i+1} is an eigenvector of S^i associated with eigenvalue 0, we have

$$g^i - g^{i+1} = (x^{i+1} - x^i)' S^i (x^{i+1} - x^i). \tag{56}$$

Assuming that the last (zero) eigenvalue of S^i has multiplicity one and that S^i does not tend to a matrix for which the last eigenvalue has multiplicity larger than one (as can be expected to hold in practice), it follows from $\text{Lim}_{i\to\infty} (g^i - g^{i+1}) = \text{Lim}_{i\to\infty}(x^{i+1} - x^i)' S^i (x^{i+1} - x^i)) = 0$ that $(x^{i+1} - x^i)$ tends to a vector proportional to the last (uniquely determinable) eigenvector of S^i. Hence $\text{Lim}_{i\to\infty} (x^{i+1} - x^i - \mu^i x^{i+1}) = 0$ for an unknown scalar μ^i. As a consequence, $\text{Lim}_{i\to\infty} ((1 - \mu^i)x^{i+1} - x^i) = 0$, and it follows from the unit length constraints on x^{i+1} and x^i that $(1 - \mu^i)^2$ tends to 1, hence that μ^i tends to 0 or 2. As a result, we have either $\text{Lim}_{i\to\infty} (x^{i+1} - x^i) = 0$, or $\text{Lim}_{i\to\infty} (-x^{i+1} - x^i) = 0$. The latter can, however, be avoided by determining eigenvectors always such that, for instance, the first nonzero element is positive.

References

de Leeuw, J. (1982). Generalized eigenvalue problems with positive semi-definite matrices. *Psychometrika, 47*, 87–93.

de Leeuw, J. & Heiser, W. (1980). Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah (Ed.), *Multivariate analysis V* (pp. 501–522). Amsterdam: North Holland.

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

Harman, H. H. (1976). *Modern factor analysis* (3rd ed.). Chicago: University of Chicago Press.

Henderson, H. V., & Searle, S. R. (1981). The Vec-permutation matrix, the Vec operator and Kronecker products: A review. *Linear and multilinear algebra, 9*, 271–288.

Kaiser, H. F., & Dickman, K. W. (1959). *Analytic determination of common factors*. Unpublished manuscript, University of Illinois.

Kiers, H. A. L. (1990). Majorization as a tool for optimizing a class of matrix functions. *Psychometrika, 55*, 417–428.

Kiers, H. A. L. (1991). Simple structure in components analysis techniques for mixtures of qualitative and quantitative variables. *Psychometrika, 56*, 197–212.

Kiers, H. A. L., & ten Berge, J. M. F. (1989). Alternating least squares algorithms for simultaneous components analysis with equal component weight matrices for all populations. *Psychometrika, 54*, 467–473.

Kiers, H. A. L., & ten Berge, J. M. F. (1992). Minimization of a class of matrix trace functions by means of refined majorization. *Psychometrika, 57*, 371–382.

Kiers, H. A. L., & ten Berge, J. M. F. (1994). Hierarchical relations between methods for simultaneous component analysis and a technique for rotation to a simultaneous structure. *British Journal of Mathematical and Statistical Psychology*, **47**, 109–126.

Kroonenberg, P. M., & de Leeuw, J. (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika, 45*, 69–97.

McDonald, R. P. (1968). A unified treatment of the weighting problem. *Psychometrika, 33*, 351–384.

McDonald, R. P., Torii, Y., & Nishisato, S. (1979). Some results on proper eigenvalues and eigenvectors with applications to scaling. *Psychometrika*, *44*, 211–227.

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

Millsap, R. E., & Meredith, W. (1988). Component analysis in cross-sectional and longitudinal data. *Psychometrika*, *53*, 123–134.

Nierop, A. F. M. (1993). *Multidimensional analysis of grouped variables: An integrated approach*. Leiden: DSWO Press.

Ostrowski, A. M. (1969). *Solutions of equations and systems of equations*. New York: Academic Press.

Tatsuoka, M. M. (1971). *Multivariate analysis: Techniques for educational and psychological research*. New York: Wiley.

ten Berge, J. M. F. (1983). A generalization of Kristof's theorem on the trace of certain matrix products. *Psychometrika*, *48*, 519–523.

ten Berge, J. M. F., Knol, D. L., & Kiers, H. A. L. (1988). A treatment of the orthomax rotation family in terms of diagonalization, and a re-examination of a singular value approach to varimax rotation. *Computational Statistics Quarterly*, *3*, 207–217.

Manuscript received 4/26/93
Final version received 4/6/94