PSYCHOMETRIKA—VOL. 66, NO. 2, 289–306
JUNE 2001

A SIMPLE GENERAL PROCEDURE FOR ORTHOGONAL ROTATION

ROBERT I. JENNRICH

UNIVERSITY OF CALIFORNIA AT LOS ANGELES

A very general algorithm for orthogonal rotation is identified. It is shown that when an algorithm parameter α is sufficiently large the algorithm converges monotonically to a stationary point of the rotation criterion from any starting value. Because a sufficiently large α is in general hard to find, a modification that does not require it is introduced. Without this requirement the modified algorithm is not only very general, but also very simple. Its implementation involves little more than computing the gradient of the rotation criterion. While the modified algorithm converges monotonically from any starting value, it is not guaranteed to converge to a stationary point. It, however, does so in all of our examples. While motivated by the rotation problem in factor analysis, the algorithms discussed may be used to optimize almost any function of a not necessarily square column-wise orthonormal matrix. A number of these more general applications are considered. Empirical examples show that the modified algorithm can be reasonably fast, but its purpose is to save an investigator's effort rather than that of his or her computer. This makes it more appropriate as a research tool than as an algorithm for established methods.

Key words: factor analysis, quartimax, varimax, orthomax, simultaneous rotation, simultaneous diagonalization, singular value decomposition, pairwise methods, global convergence.

1. Introduction

Let Λ be a factor loading matrix and let $Q(\Lambda)$ be the value of an orthogonal rotation criterion at Λ. Consider maximizing $Q(\Lambda)$ over all rotations Λ of an initial loading matrix A, that is over all

$$\Lambda = AT$$

where T is an arbitrary orthogonal matrix. Another way to say this is that we want to maximize the function

$$f(T) = Q(AT) \tag{1}$$

over all orthogonal matrices T.

More generally let f be an arbitrary function defined on a subset of p by k matrices with $p \geq k$ that contains all of the p by k column-wise orthonormal matrices $\mathcal{M}$. We wish to maximize f over $\mathcal{M}$. Consider the following simple algorithm for this purpose.

Choose $\alpha \geq 0$ and a T in $\mathcal{M}$.

(a) Compute $G = df/dT$.

(b) Find the singular value decomposition UDV' of $G + \alpha T$.

(c) Replace T by UV' and go to (a) or stop.

Here df/dT is the matrix of partial derivatives of f at T. In most matrix languages step (b) is a single line of code. We will call this very simple algorithm the basic singular value (BSV) algorithm. It may not converge to anything of interest and in fact may not even be defined, but we will show that under very general conditions it is defined, and when α is sufficiently large it

The author is indebted to Michael Browne for significant insights and advice on the rotation problem and specific comments on this manuscript, and to the review team for valuable suggestions.

Requests for reprints should be sent to Robert I. Jennrich, Department of Statistics, University of California, Los Angeles, CA 90095. E-Mail: rij@stat.ucla.edu

0033-3123/2001-2/1999-0715-A $00.75/0

is monotone, and convergent to a stationary point from any starting value. Unfortunately, finding a specific sufficiently large α may not be easy. For this reason we will show how to modify the BSV algorithm so that a specific α is not required. The resulting algorithm is still simple and monotone from any starting value, but our theory does not guarantee it will converge to a stationary point. In all our examples, however, it has never failed to do so.

A number of authors have used BSV algorithms without identifying the structure displayed above. The simultaneous varimax algorithm introduced independently by Horst (1965) and Sherin (1966) and its extension to orthomax rotation by Mulaik (1972) are BSV algorithms with $\alpha = 0$. Ten Berge's (1984) singular value decomposition algorithm for simultaneously diagonalizing a set of symmetric matrices in the least squares sense is also a BSV algorithm with $\alpha = 0$. Kiers' (1990) algorithm for optimizing a class of matrix functions is a BSV algorithm using a carefully specified positive α as is the algorithm of Kiers, ten Berge, Takane, and de Leeuw (1990) for DEDICOM rotation.

We will provide a simple general motivation for the BSV algorithm based on majorization and show that the algorithms mentioned above and others are in fact BSV algorithms. General discussions of majorization algorithms may be found in de Leeuw (1994) and Heiser (1995). We will derive some basic convergence properties of the BSV algorithm under the assumption that α is sufficiently large and show that such an α always exists. Included are some important properties not previously demonstrated, including convergence to a stationary point. As mentioned, we will show how to modify the BSV algorithm when a sufficiently large α is not known. Finally, we will look at a number of specific applications to demonstrate that the modified algorithm is simple to use, widely applicable, and can be reasonably efficient. It is, however, the simplicity and generality of the modified algorithm, rather than its efficiency, that makes it attractive.

2. Derivation of the BSV Algorithm

Given matrices A and B of the same size, let $(A, B) = \text{tr}(A'B)$. This is the Frobenius product of A and B, and $\|A\| = (A, A)^{1/2}$ is the Frobenius norm of A. Let $\mathcal{R}$ be the set of all p by k matrices and let

$$\mathcal{B} = \{X \in R : \|X\| \leq \sqrt{k}\}.$$

Since for any T in $\mathcal{M}$, $\|T\| = \sqrt{k}$, $\mathcal{B}$ contains $\mathcal{M}$. Assume f is defined and twice continuously differentiable on $\mathcal{B}$. Let T be a matrix in $\mathcal{M}$. We will view this as the current value of T and eventually update it. For any X in $\mathcal{R}$ let

$$\tilde{f}(X) = f(T) + (G, X - T) - \frac{1}{2}\alpha\|X - T\|^2 \tag{2}$$

where $G = df/dT$ is the gradient of f at T. Clearly $\tilde{f}$ and f have the same value at T. It is shown in the Appendix that the gradient of $\tilde{f}$ at T is G and hence $\tilde{f}$ and f have the same gradient at T. It is also shown in the Appendix that if f is twice continuously differentiable on $\mathcal{B}$, then there is an α such that $f \geq \tilde{f}$ on $\mathcal{B}$. Thus for α sufficiently large

$$f(\tilde{T}) \geq \tilde{f}(\tilde{T}) \tag{3}$$

for all $\tilde{T}$ in $\mathcal{M}$. Assume (3) holds for a given α and assume that for some $\tilde{T}$ in $\mathcal{M}$, $\tilde{f}(\tilde{T}) > \tilde{f}(T)$. Then

$$f(\tilde{T}) \geq \tilde{f}(\tilde{T}) > \tilde{f}(T) = f(T).$$

Thus $\tilde{f}(\tilde{T}) > \tilde{f}(T)$ implies $f(\tilde{T}) > f(T)$. In words, increasing the value of $\tilde{f}$ over its current value increases the value of f over its current value. We will show that the BSV algorithm proceeds by repeatedly maximizing $\tilde{f}$ over $\mathcal{M}$. Because under condition (3), f majorizes $\tilde{f}$ on

$\mathcal{M}$, the BSV algorithm is called a majorization algorithm when α is sufficiently large to satisfy condition (3).

We turn now to maximizing $\tilde{f}$ over $\mathcal{M}$. By expanding (2), one can show that for any $\tilde{T}$ in $\mathcal{M}$

$$\tilde{f}(\tilde{T}) = \text{const} + (G + \alpha T, \tilde{T})$$

where "const" is a constant with respect to $\tilde{T}$. Cliff (1966) has shown that for any column-wise orthonormal matrix S and any fixed matrix M of the same size, (M, S) is maximized when $S = UV'$ where UDV' is any singular value decomposition (s.v.d.) of M. Using this result, $\tilde{T} = UV'$ maximizes $\tilde{f}$ over $\mathcal{M}$ if UDV' is a s.v.d. of $G + \alpha T$. One might proceed by replacing T by $\tilde{T}$ and repeating the procedure just described until it hopefully converges. This is the BSV algorithm defined in section 1. When α is sufficiently large it is a majorization algorithm that increases, or at least does not decrease, the value of f at each step. It is this that motivates the BSV algorithm.

In the context of factor analysis rotation it is shown in the Appendix that

$$G = A' \frac{dQ}{d\Lambda}. \tag{4}$$

This result and other differentiation results we need may be obtained in a number of ways. The most elementary, but often the most tedious, is to express the function to be differentiated explicitly in terms of the components of its argument and then compute partial derivatives. A particularly nice approach is to use differentials. This is the approach used in the matrix calculus book by Magnus and Neudecker (1999). We illustrate the use of differentials in the Appendix where almost all the gradients we need are derived.

The quartimax criterion (Neuhaus & Wrigley, 1954) can be written in the form

$$Q(\Lambda) = \frac{1}{4} \sum \sum \lambda_{ir}^4 \tag{5}$$

where the λ_{ir} are the components of Λ. Differentiating gives

$$\frac{dQ}{d\Lambda} = \Lambda^3$$

where Λ^3 is the element-wise cube of Λ. Thus for quartimax rotation

$$G = A'\Lambda^3.$$

We show in section 5 that for quartimax rotation $\alpha = 0$ is sufficiently large to satisfy condition (3). The MATLAB computer code for the quartimax BSV algorithm with $\alpha = 0$ is as follows:

```
for iter = 0:5
   L = A*T;
   G = A'*(L.^3);
   [U,S,V] = svd(G);
   T = U*V';
   end
```

Clearly this is simple. To illustrate this algorithm we applied it to a random rotation of a 100 by 10 loading matrix with perfect simple structure to see if it could recover the perfect simple structure. A loading matrix has perfect simple structure when it has at most one nonzero element in each row. To show convergence, recall that for orthogonal rotation maximizing the quartimax

criterion is equivalent to minimizing the quartimin criterion (Carroll, 1953)

$$Q^*(\Lambda) = \sum\sum_{r \neq s} (\lambda_r^2, \lambda_s^2) \tag{6}$$

where λ_r^2 is the r-th column of Λ^2 the element-wise square of Λ. The quartimin criterion has the property that $Q^*(\Lambda) = 0$ if and only if Λ has perfect simple structure. Thus monitoring the quartimin value allows one to view convergence to perfect simple structure. Inserting code into the program above to output the value of $Q^*(\Lambda)$ at each iteration gave:

Iter	Quartimin
0	77.3442
1	56.9363
2	26.2816
3	5.8018
4	0.0585
5	0.0000

On this example, the BSV algorithm with $\alpha = 0$ converged to a perfect simple structure solution in 5 iterations. Assume now that an investigator wonders what would happen if the λ_{ir}^4 in the quartimax criterion were replaced by $|\lambda_{ir}|^3$. That is, what would happen if $Q(\Lambda)$ for the quartimax criterion were replaced by

$$Q(\Lambda) = \frac{1}{3}\sum\sum |\lambda_{ir}|^3$$

Assuming it has not already been named, we will call this the cubimax criterion. For cubimax

$$\frac{dQ}{d\Lambda} = \Lambda^2 \cdot \text{sign}(\Lambda)$$

where $\text{sign}(\Lambda)$ is Λ with its components replaced by their signs $(-1, 0, 1)$ and "$\cdot$" denotes the element-wise product. Replacing the formula for G in the computer code above by

$$G = A'(\Lambda^2 \cdot \text{sign}(\Lambda))$$

gives a BSV algorithm for cubimax rotation. When applied to the quartimax data used above it converged in 5 iterations to a Λ with perfect simple structure. Apparently maximizing the cubimax criterion also recovers perfect simple structure when it exists. This can be proved analytically, but for the author it was first discovered empirically. One might also want to compare cubimax to varimax (Kaiser, 1958). Again, all that is required is finding a formula for the gradient of the varimax criterion and changing one line of code. This is done in section 6. Clearly this approach makes very efficient use of an investigator's time if not that of his or her computer.

3. Convergence of the BSV Algorithm

We would like the iterates T of the BSV algorithm to converge to a maximizer or at least to a stationary point of f restricted to $\mathcal{M}$.

The strongest evidence in support of some form of convergence to date is that, as indicated, specific forms of the BSV algorithm have been used successfully by a number of authors.

On the theoretical side, Sherin (1966) has observed that a fixed point of his algorithm, that is an orthogonal matrix T that doesn't change when his algorithm is applied to it, is a stationary point of f. Ten Berge, Knol, and Kiers (1988) have shown that for the problem of simultaneously diagonalizing a set of symmetric matrices, the singular value decomposition algorithm of ten Berge (1984) is monotone provided the matrices are nonnegative definite. Here monotone means only that the algorithm is nondecreasing, that is $f(\tilde{T}) \geq f(T)$ where $\tilde{T}$ is the update of T

produced by the algorithm. Kiers (1990) has shown that his algorithm for optimizing a class of matrix functions is also monotone as have Kiers, ten Berge, Takane, and de Leeuw (1990) for their DEDICOM algorithm. We will show that these results hold for any BSV algorithm when α is sufficiently large and we will provide some important additional results.

It is shown in the Appendix that if g is a scalar valued function defined and differentiable on $\mathcal{B}$, then T in $\mathcal{M}$ is a stationary point of g restricted to $\mathcal{M}$ if and only if for some symmetric matrix S

$$G = TS \tag{7}$$

where $G = dg/dT$ is the gradient of g at T. The following is a basic theorem for our development.

Theorem 1. If α satisfies condition (3) and T is not a stationary point of f restricted to $\mathcal{M}$, then

$$f(\tilde{T}) > f(T)$$

where $\tilde{T} = UV'$ is the update defined by the BSV algorithm.

Proof. Since T is not a stationary point of f restricted to $\mathcal{M}$, it follows from (7) and the fact that f and $\tilde{f}$ have the same gradient at T, that T is not a stationary point of $\tilde{f}$ restricted to $\mathcal{M}$. Thus T cannot maximize $\tilde{f}$ restricted to $\mathcal{M}$ because if it did, it would have to be a stationary point of $\tilde{f}$ restricted to $\mathcal{M}$ (Kaplan, 1999, p. 97). Since $\tilde{T}$ maximizes $\tilde{f}$ restricted to $\mathcal{M}$, $\tilde{f}(\tilde{T}) > \tilde{f}(T)$. Using (3) and the fact that f and $\tilde{f}$ have the same value at T

$$f(\tilde{T}) \geq \tilde{f}(\tilde{T}) > \tilde{f}(T) = f(T)$$

which completes the proof. □

There are a number of important consequences of Theorem 1. If α is sufficiently large to satisfy condition (3), it implies that the BSV algorithm is monotone and its fixed points are stationary points. Note also that $f(\tilde{T}) = f(T)$ implies T is a stationary point. Under the assumption that the mapping $T \to \tilde{T}$ defined by the BSV algorithm is continuous, it follows from Theorem 1 and the Zangwill global convergence theory (see, e.g., Luenberger, 1984, p. 183) that independent of the initial value of T, any limit point of the iterates produced by the BSV algorithm will be a stationary point of f restricted to $\mathcal{M}$. We will show in the Appendix that this is true even if the mapping $T \to \tilde{T}$ is not continuous. This is global convergence in precisely the same sense as that enjoyed by the well known EM algorithm (Dempster et al., 1977) which is also a majorization algorithm (Heiser, 1995) and arguably the best known majorization algorithm. Because $\mathcal{M}$ is closed and bounded, the iterates produced by the BSV algorithm must have at least one limit point. In practice there is only one and the iterates converge to it. If there is more than one, all such limit points are stationary points. In either case we will simply say the iterates converge to a stationary point. The term global convergence used in conjunction with the Zangwill global convergence theory is a bit misleading. It does not mean that the iterates converge to a global maximizer of f restricted to $\mathcal{M}$, but rather that the convergence is global with respect to the starting value. That is, from any starting value the iterates converge to a stationary point of f restricted to $\mathcal{M}$.

The importance of Theorem 1 is that knowing only that a fixed point of an algorithm is a stationary point or knowing only that the algorithm is nondecreasing does not imply any of the other properties of the previous paragraph. For example, knowing only that an algorithm is monotone does not imply that it converges to anything of interest. Its limit need not even be a stationary point of f restricted to $\mathcal{M}$. Thus Theorem 1 significantly extends the theoretical results

of the authors mentioned in the third paragraph of this section. Most importantly it implies that their algorithms are not only monotone, but actually converge to stationary points.

To measure the speed of a BSV algorithm we need a stopping rule. For this we need a simple way to identify stationary points. It is shown in the appendix that T is a stationary point of f restricted to $\mathcal{M}$ if and only if

$$v = \|\text{skm}(T'G)\| + \|(I - TT')G\| \tag{8}$$

is zero. Here G is the gradient of f at T, $\text{skm}(M) = \frac{1}{2}(M - M')$ denotes the skew-symmetric part of a square matrix M, and I is an identity matrix. Note that the second term on the right in (8) is zero when T is an orthogonal matrix. We will use v as an index to measure convergence and stop when $v < 10^{-6}$.

One can generalize the BSV algorithm a bit by letting α be a function of T. Another generalization is to replace αT in step (b) by TW where W is a positive definite matrix or in the case when T is square by WT. In both cases W may be a function of T. As before, if the resulting algorithms are majorization algorithms they are globally convergent to stationary points. This is shown in the appendix. Kiers and ten Berge (1992) show that for their algorithm an appropriate choice of W can improve the speed of the resulting BSV algorithm.

Note from (2) that as $\alpha \to \infty$ the step $\tilde{T} - T \to 0$. Thus using values of α larger than necessary may slow the BSV algorithm and indeed seems to do so in one of our examples.

4. Removing the Requirement that α Be Sufficiently Large

A serious problem with using the BSV algorithm as a majorization algorithm is the necessity to find a specific α that, in fact, makes it a majorization algorithm. This usually requires a detailed study of the specific rotation criterion used and may involve a substantial theoretical effort. While this seems appropriate for well established rotation criteria, it is a definite handicap in developing and studying new ones. Moreover, each new criterion requires finding a new α, and it is not at all clear how to find an appropriate α in general. Thus, used as a majorization algorithm, the BSV algorithm is neither simple nor general. One may, however, get around the appropriate α problem by dropping the requirement that the BSV algorithm be used only as a majorization algorithm.

For example, one might simply run the BSV algorithm with an arbitrary α and starting value T. If it converges monotonically to a stationary point, one is little motivated to search out an α whose use will guarantee monotone convergence to a stationary point, at least not for the problem at hand.

Alternatively, one might use the following modification of the BSV algorithm. Given any real α, any $\Delta\alpha > 0$, and any starting value T in $\mathcal{M}$:

(a) Run one step of the BSV algorithm using the current values of α and T.

(b) If the step in (a) increases the value of f, replace T by its update and go to (a).

(c) If the step in (a) does not increase the value of f, replace α by $\alpha + \Delta\alpha$ and go to (a).

This algorithm is clearly monotone. If during the iterations α becomes large enough to make this modified BSV algorithm a majorization algorithm, it must converge to a stationary point. Also, if it reaches a point after which it fails to move, it must be at a stationary point. Otherwise α will eventually become large enough to make it a majorization algorithm, at which point it must move in step (b). Convergence to a stationary point, however, is not guaranteed by our theory. Nevertheless, in all of our examples this has never failed to happen. This is a common situation. The Fisher scoring and Newton algorithms are often modified to make them monotone, but the modifications used do not guarantee convergence to a stationary point. These modified algorithms, however, usually converge to stationary points and are extensively used.

The modification of the BSV algorithm described depends on two parameters α and $\Delta\alpha$. Their choice can effect the speed of the modified algorithm. In our examples we had no difficulty finding satisfactory values for these parameters.

5. The BSV Algorithm for Quartimax Rotation

We begin by showing that for quartimax rotation $\alpha = 0$ is sufficient to satisfy condition (3). It is shown in the Appendix that if f is convex on $\mathcal{B}$, then

$$\tilde{f} \geq f$$

on $\mathcal{B}$ when $\alpha = 0$. Hence it is sufficient to show f is convex on $\mathcal{B}$. To this end note that λ_{ir}^4 is a convex function of λ_{ir}. Since λ_{ir} is a linear function of Λ, λ_{ir}^4 is a convex function of Λ. Using (5) the quartimax criterion $Q(\Lambda)$ is a convex function of Λ because it is a sum of convex functions of Λ. Since AX is a linear function of X,

$$f(X) = Q(AX)$$

is a convex function of X for all X in $\mathcal{R}$.

To demonstrate the convergence of the BSV algorithm and show that it is not painfully slow, we compared it to the standard pairwise (PW) algorithm (Neuhaus and Wrigley, 1954). Both were applied to a number of randomly generated initial loading matrices. Because for the most part efficiency is important only for large loading matrices, we consider only the case of 100 by 10 matrices. Let

$$\Lambda_0 = I \otimes u \tag{9}$$

where I is a 10 by 10 identity matrix, u is a 10 component column vector of ones, and $\otimes$ denotes the Kronecker product. Then Λ_0 is a 100 by 10 loading matrix with perfect simple structure. We will consider initial loading matrices that are random rotations of Λ_0.

Let Z be a 10 by 10 random matrix whose components are independent standard normal variables and let QR be a QR factorization (Golub & van Loan, 1989) of Z. Then Q is a random orthogonal matrix, that is one that is uniformily distributed over the group of orthogonal 10 by 10 matrices and

$$A = \Lambda_0 Q$$

is a random rotation of Λ_0. Using this method 100 loading matrices were generated and rotated using the BSV and PW algorithms. Using (8) we declared these algorithms converged when

$$\|\text{skm}(T'G)\| < 10^{-6}. \tag{10}$$

When this happens the solutions obtained are to a good approximation stationary points of the quartimax criterion. This happened in every case for both algorithms. Moreover, both algorithms converged to perfect simple structure solutions in every case. To measure speed the number of floating point operations (FLOPS) in multiples of 10,000 required for convergence were recorded. The mean number of FLOPS required are given in Table 1 under the heading Problem 1.

On this problem the BSV and PW algorithms have about the same speed.

To see what happens when Λ_0 does not have perfect structure, Λ_0 was replaced by a contaminated form

$$\Lambda_0^* = \Lambda_0 + .2\,U$$

where U is a 100 by 10 matrix of ones. Using 100 random rotations of Λ_0^* both the BSV and PW quartimax algorithms converged to a stationary point in every case and in every case their

TABLE 1.
The average number of FLOPS in multiples of 10,000 for convergence of the quartimax BSV and PW algorithms for the perfect simple structure case (Problem 1), the less than perfect simple structure case (Problem 2), and the unstructured case (Problem 3).

	Problem 1	Problem 2	Problem 3
BSV	46.88	163.47	2844
PW	53.18	102.18	896

function values f differed by less than 10^{-12}. The mean number of FLOPS required are given in Table 1 under the heading Problem 2. For this problem the PW algortihm was about 50% faster than the BSV algorithm.

Using a totally random initial loading matrix both algorithms are much slower. Using 100 initial loading matrices whose components were realizations of independent standard normal variables, both algorithms converged to stationary points in every case. In 89 cases the function values f differed by less than 10^{-12} and in the other cases the difference exceeded 0.4. Among these the function value for the BSV algorithm exceeded that for the PW algorithm 8 out of 11 times. The average number of FLOPS in multiples of 10,000 required for convergence is given in Table 1 under the heading Problem 3. For this purely random loading matrix problem the PW algorithm is about 3 times faster than the BSV algorithm.

In terms of the computing time required, the BSV algorithm is reasonably fast. In the totally random case it required roughly 30 million FLOPS. This is about 3 seconds on current desk top computers. Problems with structure require only a fraction of a second.

The purpose of this section was to show that for quartimax rotation the BSV algorithm is not too slow to be useful, that its solutions are in fact stationary points, and that the function values produced are comparable to those produced by the PW algorithm.

6. BSV Orthomax Algorithms

The quartimax criterion is one of the orthomax family of criteria (Harman, 1960). The orthomax family may be written in the form

$$Q(\Lambda) = (\Lambda^2, \Lambda^2 - \gamma\overline{\Lambda^2})/4 \tag{11}$$

where $\overline{\Lambda^2}$ is Λ^2 with each element replaced by the mean of the corresponding column of Λ^2. When $\gamma = 0$, $Q(\Lambda)$ is the quartimax criterion and when $\gamma = 1$ it is the varimax criterion.

It is shown in the appendix that the gradient of the orthomax criterion is

$$\frac{dQ}{d\Lambda} = \Lambda \cdot (\Lambda^2 - \gamma\overline{\Lambda^2})$$

which again is easy to evaluate. When $\gamma > 0$, f is no longer convex, and we cannot use the argument of the previous section to choose $\alpha = 0$. We will nevertheless use this value because the BSV orthomax algorithm with $\alpha = 0$ has never failed to converge to a stationary point on our trials.

Using the 300 initial loading matrices from the previous section we compared the BSV algorithm for varimax rotation with the PW varimax algorithm (Kaiser, 1958). As in the previous section the BSV and PW algorithms converged to stationary points in every case. For the perfect simple structure examples and those with structure less than perfect the converged function values for the BSV and PW algorithms differed by less than 10^{-12} in every case. In the totally unstructured case the function values differed by less than 10^{-10} on 86 trials and exceeded .01 on the others. In the latter the value for the BSV algorithm exceeded that for the PW algorithm on 5

TABLE 2.
The average number of FLOPS in multiples of 10,000 for convergence of the varimax BSV and PW algorithms for the perfect simple structure case (Problem 1), the less than perfect simple structure case (Problem 2), and the unstructured case (Problem 3).

	Problem 1	Problem 2	Problem 3
BSV	97.80	82.37	2919
PW	76.83	194.76	1388

out of 14 trials. The average number of FLOPS in multiples of 10,000 for the two algorithms are given in Table 2.

The speeds for varimax rotation are roughly what we saw for quartimax rotation. They demonstrate, we believe, that for varimax rotation the BSV algorithm is reasonably efficient at least on the examples considered. Note, we do not claim $\alpha = 0$ is sufficiently large to satisfy condition (3). In every case considered, however, using $\alpha = 0$ gave convergence to a stationary point and the same function values for all of the examples with structure.

7. Simultaneous Diagonalization

A rotation problem of a somewhat different character is that of finding an orthogonal matrix T to simultaneously diagonalize a set $E_1, \cdots, E_k$ of symmetric p by p matrices in the least squares sense. This problem arises in multidimensional scaling (de Leeuw & Pruzansky, 1978). More specifically the problem is to minimize

$$\sum_{i=1}^{k} \|\text{ndg}(T'E_iT)\|^2 \tag{12}$$

over all p by p orthogonal matrices T. Here $\text{ndg}(M)$ denotes the nondiagonal part of M. Minimizing (12) is equivalent to maximizing

$$f(T) = \sum_{i=1}^{k} \|\text{dg}(T'E_iT)\|^2/4 \tag{13}$$

over all p by p orthogonal matrices T. Here $\text{dg}(M)$ denotes the diagonal part of M.

It is shown in the Appendix that

$$G = \sum_{i=1}^{k} E_iT\,\text{dg}(T'E_iT)$$

is the gradient of f at T and this together with a choice of α defines the BSV algorithm for this problem.

When $\alpha = 0$, the BSV algorithm is the singular value decomposition algorithm of ten Berge (1984, Eq. (24)). As noted, ten Berge, Knol, and Kiers (1988) have shown that when the E_i are nonnegative definite (n.n.d.), ten Berge's singular value decomposition algorithm is monotone. We can assert more, namely that it is also globally convergent to a stationary point of f restricted to $\mathcal{M}$, the manifold of all p by p orthogonal matrices.

To do this we must show that condition (3) is satisfied. As in section 5, it is sufficient to show $f(X)$ is a convex function of X for X in $\mathcal{B}$. Let x_r be the r-th column of X. Then

$$f(X) = \sum_i \sum_r (x_r'E_ix_r)^2. \tag{14}$$

Note that since E_i is n.n.d. $x_r'E_ix_r$ is a convex function of x_r and hence of X. Note also that $x_r'E_ix_r \geq 0$. Since $(x_r'E_ix_r)^2$ is an increasing convex function of $x_r'E_ix_r$, $(x_r'E_ix_r)^2$ is a convex

function of X. It follows from (14) that $f(X)$ is a convex function of X because it is a sum of convex functions of X.

Ten Berge, Knol, and Kiers (1988) show that ten Berge's algorithm may be used with indefinite E_i by modifying the criterion (12). This is based on the observation that one may replace each E_i in (12) by a n.n.d. matrix $E_i + \lambda_i I$ without changing its value at an orthogonal matrix T. Thus an indefinite E_i problem may be replaced by one with n.n.d. E_i. With this modification, ten Berge's algorithm is again a BSV algorithm with $\alpha = 0$ and is monotone and globally convergent to a stationary point of f restricted to $\mathcal{M}$.

8. Optimizing a Class of Matrix Functions

Kiers (1990) introduced a majorization algorithm for optimizing a class of matrix functions useful in a variety of statistical applications. We will show his rotation algorithm is a BSV algorithm.

For rotation problems, the functions Kiers considers are of the form

$$f(X) = c + \operatorname{tr} AX + \sum_{j=1}^{m} \operatorname{tr} B_j X C_j X' \tag{15}$$

where X is any matrix in $\mathcal{R}$, c is a given constant, and A, B_j, C_j are given matrices of appropriate sizes. As shown in the Appendix, the gradient of f at X is

$$G = A' + \sum B_j X C_j + \sum B_j' X C_j'. \tag{16}$$

Kiers considers minimizing rather than maximizing f restricted to $\mathcal{M}$. Given T in $\mathcal{M}$, its update using Kiers' algorithm is $\tilde{T} = UV'$ where UDV' is the s.v.d. of

$$T - \frac{1}{2\sum \alpha_j}\left(A' + \sum B_j T C_j + \sum B_j' T C_j'\right) \tag{17}$$

where the $\alpha_j \geq 0$ are defined by Kiers. Let $\alpha = 2\sum \alpha_j$. Using (16)

$$T - \frac{1}{\alpha} G = UDV'.$$

It follows that

$$-G + \alpha T = U(\alpha D)V'.$$

Since αD is a nonnegative diagonal matrix and $-G$ is the gradient of $-f$ at T, the update $\tilde{T} = UV'$ used by Kiers is the update used by the BSV algorithm to maximize $-f$ restricted to $\mathcal{M}$ with $\alpha = 2\sum \alpha_j$. Hence Kiers' algorithm is a BSV algorithm. Kiers shows that his algorithm is monotone. But he also shows that when $\alpha = 2\sum \alpha_j$, condition (3) is satisfied for $-f$ and hence Kiers' algorithm is not only monotone, but also globally convergent to a stationary point.

Kiers and ten Berge (1992) present a modification to Kiers' algorithm that is designed to converge faster. Kiers and ten Berge assume that some of the C_j in (15) are n.n.d. and that these are the first k of the C_j. Given T in $\mathcal{M}$ the new algorithm uses the update $\tilde{T} = UV'$ where UDV' is the s.v.d. of

$$\sum_{j=1}^{m} F_j' - A' \tag{18}$$

where

$$F_j' = \begin{cases} 2\rho_j TC_j - (B_j + B_j')TC_j & j \le k \\ 2\lambda_j T - B_j TC_j - B_j' TC_j' & j > k \end{cases}$$

and ρ_j and λ_j are nonnegative values specified by Kiers and ten Berge. Using (16) and some algebra, (18) can be expressed in the form

$$-G + TW$$

where

$$W = \sum_{j=1}^{k} 2\rho_j C_j + \sum_{j=k+1}^{m} 2\lambda_j I.$$

Thus the Kiers and ten Berge algorithm is the BSV algorithm for maximizing $-f$ with αT in Step (b) replaced by TW. Kiers and ten Berge show their algorithm is monotone, but as with Kiers' algorithm it follows from our general theory that it is also globally convergent to a stationary point.

9. Fitting the DEDICOM Model

Kiers, ten Berge, Takane, and de Leeuw (1990) give a method for least squares fitting of the DEDICOM model. They show that this may be reduced to finding a T in $\mathcal{M}$ that maximizes $\|T'YT\|^2$ where Y is the data matrix to be fitted. For all X in $\mathcal{R}$ let

$$f(X) = \|X'YX\|^2/2. \tag{19}$$

The fitting problem may be viewed as that of maximizing f restricted to $\mathcal{M}$. Kiers et al. update T in $\mathcal{M}$ by any $\tilde{T}$ whose columns are an orthonormal basis of the column space of

$$YTT'Y'T + Y'TT'YT + 2\tilde{\alpha}T$$

where $\tilde{\alpha}$ is no less than the largest eigenvalue of the symmetric part of $-Y \otimes T'YT$. As shown in the Appendix, the gradient of f at T is

$$G = YTT'Y'T + Y'TT'YT,$$

and hence the update $\tilde{T}$ is any matrix whose columns are an orthonormal basis of the column space of $G + 2\tilde{\alpha}T$. One such $\tilde{T}$ is given by $\tilde{T} = UV'$ where UDV' is a s.v.d. of $G + 2\tilde{\alpha}T$. Thus the Kiers et al. algorithm is a BSV algorithm with $\alpha = 2\tilde{\alpha}$. The value of $\tilde{\alpha}$ may change from iteration to iteration. Kiers et al. also consider using an $\tilde{\alpha}$ that does not change. One such $\tilde{\alpha}$ is the largest singular value of $Y \otimes Y$ which is the square of the largest singular value of Y.

Kiers et al. show that their algorithm is monotone. They do not show that condition (3) is satisfied, but this follows fairly easily from their results and hence their algorithm is not only monotone, but also globally convergent to a stationary point of f restricted to $\mathcal{M}$.

When $\tilde{\alpha} = 0$ the Kiers et al. algorithm is the algorithm of Takane (1985). Hence Takane's algorithm is a BSV algorithm with $\alpha = 0$. Kiers et al. show that Takane's algorithm need not be monotonic. They found, however, that this did not happen on their noncontrived examples when using good starting values. Moreover, on these examples Takane's algorithm was very efficient compared to the Kiers et al. algorithm. This shows that using values of α larger than necessary may significantly slow the BSV algorithm.

To expand on this point we compared the modified BSV algorithm of section 4, which unlike Takane's algorithm is monotone, to the algorithm of Kiers et al. We used 100, 10 by 10 data matrices Y whose components were realizations of independent standard normal variables.

The matrices T were 10 by 5 with initial value

$$T = \begin{pmatrix} I \\ 0 \end{pmatrix}$$

where I is an identity matrix. For the modified BSV algorithm, $\alpha = 0$ and $\Delta\alpha = .1$ were used, and for the Kiers et al. algorithm $\tilde{\alpha}$ was equal to the square of the largest singular value of Y.

Both algorithms converged to a stationary point on all 100 trials. In terms of total FLOPS used, the modified BSV algorithm was 4.09 times faster than the Kiers et al. algorithm. In 91 of the trials the values of f at convergence agreed to 12 significant digits. In two trials they agreed to 5 and 7 significant digits. In both of these the modified BSV algorithm had the larger value of f. In the other 7 trials, they agreed to at most 2 significant digits and among these the modified BSV algorithm had the larger value of f on 3 trials. Clearly neither algorithm always produces a global maximizer. What we have shown is that the modified BSV algorithm converged to a stationary point in every case and with reasonable speed.

10. Discussion

The BSV algorithm unifies the theory of singular value rotation algorithms because many previous algorithms can be expressed simply as BSV algorithms, and because it provides a singular value algorithm for essentially any rotation problem.

We have shown that with an appropriate choice of α, the BSV algorithm is monotone and globally convergent to a stationary point. Moreover, we have shown that such an α always exists. We have also shown how to identify stationary points for a general rotation problem in a simple way.

Finding an appropriate α, however, can be difficult. For this reason we have shown how to modify the BSV algorithm so the difficult to find α is not required. The modification is guaranteed to be monotone, but not necessarily convergent to stationary point. Nevertheless in all of our examples it has never failed to converge to a stationary point. What makes the modification work is the fact that an α large enough to satisify condition (3) always exists as we have shown. The modified BSV algorithm is very easy to derive requiring little more than a formula for the gradient of f.

The BSV algorithm and its modification will probably be slower than special purpose algorithms such as pairwise algorithms when they exist. But our examples show that the modified BSV algorithm is reasonably efficient in several applications. Because it tends to be easy to implement, its efficiency in a given context is usually easy to evaluate.

As noted in section 3 the size of the steps produced by the BSV algorithm tend to decrease as α increases. This will lead to unnecessarily slow convergence when α is too large. A reviewer has pointed out this suggests the modified BSV algorithm may generally be faster than the BSV algorithm with α large enough to guarantee it is a majorization algorithm and this, indeed, happened with the DEDICOM example in section 9.

Although the BSV algorithm was reasonably fast in our examples, one should not conclude this will be true for all applications. We know from our experence with the EM algorithm that majorization algorithms can be painfully slow (see, e.g., Jamshidian and Jennrich, 1977) and it seems likely that this will be true for the BSV algorithm as well. It will depend on the application.

11. Appendix

11.1. Differentials and Matrix Calculus

Differentials are discussed in most advanced calculus texts, for example, Buck (1956) and Loomis and Sternberg (1968). A book length treatment of differentials in the context of matrix

calculus is given by Magnus and Neudecker (1999). We will show how to use differentials to compute gradients for functions of interest to us.

Let df_X denote the differential of a function f at X. The differential df_X is a linear function whose value at dX is denoted by $df_X(dX)$. We will use the following. If f is scalar valued then

$$df_X(dX) = (M, dX)$$

for all dX, if and only if, $M = df/dX$, the gradient of f at X.

As an example of using differentials, consider finding the gradient of f at X when

$$f(X) = Q(AX).$$

Let $\Lambda = AX$. Using the chain rule

$$\begin{aligned} df_X(dX) &= dQ_{AX}(AdX) = dQ_\Lambda(AdX) \\ &= \left(\frac{dQ}{d\Lambda}, AdX\right) = \left(A'\frac{dQ}{d\Lambda}, dX\right). \end{aligned}$$

Thus

$$\frac{df}{dX} = A'\frac{dQ}{d\Lambda}.$$

As another example, consider finding the gradient of $\tilde{f}$ given by (2) at T. Equation (2) may be written in the form

$$\tilde{f}(X) = f(T) + (G, X - T) - \frac{\alpha}{2}(X - T, X - T).$$

Differentiating gives

$$\begin{aligned} d\tilde{f}_X(dX) &= (G, dX) - \alpha(X - T, dX) \\ &= (G - \alpha(X - T), dX). \end{aligned}$$

Thus

$$\frac{d\tilde{f}}{dX} = G - \alpha(X - T)$$

which at $X = T$ is G.

As another example, consider finding the gradient of the DEDICOM criterion (19). For this

$$\begin{aligned} df_X(dX) &= (X'YX, dX'YX + X'YdX) \\ &= (X'Y'X, X'Y'dX) + (X'YX, X'YdX) \\ &= (YXX'Y'X + Y'XX'YX, dX). \end{aligned}$$

Thus the gradient of f at X is

$$YXX'Y'X + Y'XX'YX.$$

As another example, consider finding the gradient for Kiers' criterion (15). This may be expressed in the form

$$f(X) = c + (A', X) + \sum(B_jX, XC_j').$$

Viewing (B_jX, XC_j') as a function of X its differential is given by

$$\begin{aligned} d(B_jX, XC_j') &= (B_j dX, XC_j') + (B_jX, dXC_j') \\ &= (dX, B_j'XC_j') + (B_jXC_j, dX) \\ &= (B_jXC_j + B_j'XC_j', dX). \end{aligned}$$

Thus

$$df_X(dX) = (A', dX) + \left(\sum (B_jXC_j + B_j'XC_j'), dX\right),$$

and the gradient of f at X is

$$A' + \sum (B_jXC_j + B_j'XC_j').$$

As another example, consider finding the gradient of the simultaneous diagonalization criterion (13). This can be written in the form

$$f(X) = \sum_{i=1}^{k} (dg(X'E_iX), dg(X'E_iX))/4.$$

Thus

$$\begin{aligned} df_X(dX) &= \sum (dg(X'E_iX), dg(X'E_i dX)) \\ &= \sum (E_iX dg(X'E_iX), dX). \end{aligned}$$

Hence the gradient of f at X is

$$\sum E_iX dg(X'E_iX).$$

As a final example, consider finding the gradient of the orthomax criterion. Let C be the p by p matrix with all of its components equal to $1/p$. Note that C is idempotent, that is $CC = C$. For any p component column vector x

$$Cx = \begin{pmatrix} \bar{x} \\ \vdots \\ \bar{x} \end{pmatrix}$$

where $\bar{x}$ is the mean of the components of x. Using C the orthomax criterion (11) can be written in the form

$$Q(\Lambda) = (\Lambda^2, (I - \gamma C)\Lambda^2)/4.$$

Thus

$$\begin{aligned} dQ_\Lambda(d\Lambda) &= \frac{1}{2}(\Lambda \cdot d\Lambda, (I - \gamma C)\Lambda^2) + \frac{1}{2}(\Lambda^2, (I - \gamma C)(\Lambda \cdot d\Lambda)) \\ &= (\Lambda \cdot d\Lambda, (I - \gamma C)\Lambda^2)) \\ &= (d\Lambda, \Lambda \cdot ((I - \gamma C)\Lambda^2)). \end{aligned}$$

Hence

$$\frac{dQ}{d\Lambda} = \Lambda \cdot ((I - \gamma C)\Lambda^2) = \Lambda \cdot (\Lambda^2 - \gamma\overline{\Lambda^2}).$$

11.2. Stationary Points

Let g be a scalar-valued function defined and differentiable on $\mathcal{B}$. A matrix T in $\mathcal{M}$ is a stationary point of g restricted to $\mathcal{M}$ if and only if it satisfies the first order Kuhn-Tucker condition (Kaplan, 1999, p. 97). That is, if and only if the gradient of the Lagrangian

$$\ell(X) = g(X) + (S, X'X - I)$$

is zero at T for some symmetric matrix S. Differentiating

$$\begin{aligned} d\ell_T(dX) &= dg_T(dX) + 2(S, T'dX) \\ &= (G + 2TS, dX) \end{aligned}$$

where G is the gradient of g at T. Thus the gradient of ℓ at T is

$$G + 2TS,$$

and T is a stationary point if and only if

$$G = TS \tag{20}$$

for some symmetric matrix S that may differ from that in the previous equation.

We wish to express condition (20) without using S. Note that the following statements are equivalent

$$\begin{aligned} &G = TS \\ &G = TT'G, \quad T'G = S \\ &(I - TT')G = 0, \quad \text{skm}(T'G) = 0 \end{aligned}$$

Thus T is a stationary point if and only if

$$v = \|(I - TT')G\| + \|\text{skm}(T'G)\|$$

is zero.

11.3. Showing $f \geq \tilde{f}$ on $\mathcal{B}$ when α Is Sufficiently Large

We will show that when f is twice continuously differentiable on $\mathcal{B}$, there exists an α such that $f \geq \tilde{f}$ on $\mathcal{B}$.

Let $H(X)$ denote the Hessian of f at X. This is a pk by pk matrix. Because $\|H(X)\|$ is a continuous function of X for all X in $\mathcal{B}$ and because $\mathcal{B}$ is closed and bounded, $\|H(X)\|$ is bounded above for all X in $\mathcal{B}$ by some constant, say γ. Recall that for any matrix product AB, $\|AB\| \leq \|A\|\|B\|$. Thus

$$|u'H(X)u| \leq \|H(X)\|\|u\|^2 \leq \gamma\|u\|^2 \tag{21}$$

for all X in $\mathcal{B}$.

Differentiating $\tilde{f}$ gives

$$\frac{\partial^2 \tilde{f}}{\partial x_{ir}\partial x_{js}} = -\alpha\delta_{ij}\delta_{rs}$$

where δ_{ij} is one when $i = j$ and zero otherwise. This defines the Hessian $\tilde{H}$ of $\tilde{f}$ at X. More precisely $\tilde{H} = -\alpha I$ where I is a pk by pk identity matrix. Then

$$u'\tilde{H}u = -\alpha\|u\|^2.$$

It follows from (21) that $u'H(X)u \geq -\gamma\|u\|^2$ and hence

$$u'(H(X) - \tilde{H})u \geq (\alpha - \gamma)\|u\|^2.$$

Thus when $\alpha \geq \gamma$ the Hessian of $f - \tilde{f}$ is n.n.d. on $\mathcal{B}$ and hence $f - \tilde{f}$ is convex on $\mathcal{B}$. Since f and $\tilde{f}$ have the same gradient at T, $f - \tilde{f}$ has gradient zero at T and hence T minimizes $f - \tilde{f}$ over $\mathcal{B}$. Thus

$$f - \tilde{f} \geq f(T) - \tilde{f}(T) = 0$$

on $\mathcal{B}$ because f and $\tilde{f}$ have the same value at T. Hence $f \geq \tilde{f}$ on $\mathcal{B}$ when α is sufficiently large.

11.4. Showing $\alpha = 0$ Is Sufficiently Large when f Is Convex

We will show that when f is convex and differentiable on $\mathcal{B}$, $\alpha = 0$ is sufficient to satisfy condition (3).

When $\alpha = 0$, $\tilde{f}$ is linear and hence $f - \tilde{f}$ is convex. Since $f - \tilde{f}$ has gradient zero at T, T minimizes $f - \tilde{f}$ over $\mathcal{B}$. Thus

$$f - \tilde{f} \geq f(T) - \tilde{f}(T) = 0$$

on $\mathcal{B}$. Thus $f \geq \tilde{f}$ on $\mathcal{B}$ and condition (3) is satisfied when $\alpha = 0$.

11.5. Global Convergence without Assuming the Basic Algorithm Is Continuous

Let g be the function used in the BSV algorithm to map $G + \alpha T$ into the triplet (U,D,V) that defines its s.v.d. UDV'. In this notation

$$(U, D, V) = g(\nabla f(T) + \alpha T)$$

where $\nabla f(T)$ is the gradient of f at T. If g is continuous, the left-hand side is a continuous function of T. Moreover, the function that defines the BSV algorithm, that is the function that maps T into UV', is continuous. As noted above, when the function that defines the BSV algorithm is continuous and α is sufficiently large to satisfy condition (3), the basic algorithm is globally convergent. While it is tempting to believe that any algorithm one's computer uses to compute values of g is continuous, one generally doesn't actually know this, and in fact such an algorithm may not exist. Fortunately, for global convergence one does not need g to be continuous, as we will show in this section.

For each M in $\mathcal{R}$ let $\bar{g}(M)$ be the set of all T such that $T = UV'$ and UDV' is a s.v.d. of M. The set $\bar{g}(M)$ may contain more than one column-wise orthonormal matrix (e.g., $\bar{g}(0)$ contains all such matrices) and hence $\bar{g}$ must be viewed as a point to set mapping. We begin by showing:

Theorem 2. The mapping $\bar{g}$ is closed on $\mathcal{M}$.

Proof. Using the definition of a closed point to set mapping as given by Luenberger (1984, p. 185), it is sufficient to show that if

$$(M_n, T_n) \to (M, T)$$

where $M_n \in \mathcal{R}$, $T_n = U_n V_n'$, and $U_n D_n V_n'$ is a s.v.d. of M_n, then there is a s.v.d. UDV' of M such that $T = UV'$. To show this, note that because the sequences U_n, D_n, and V_n are bounded, each

has at least one limit point, say U, D, and V respectively. Let $\tilde{U}_n$, $\tilde{D}_n$, and $\tilde{V}_n$ be subsequences of U_n, D_n, V_n that converge to U, D, and V respectively and let

$$\tilde{T}_n = \tilde{U}_n \tilde{V}_n'$$
$$\tilde{M}_n = \tilde{U}_n \tilde{D}_n \tilde{V}_n'.$$

Because $\tilde{T}_n$ and $\tilde{M}_n$ are subsequences of T_n and M_n respectively, passing to the limit gives

$$T = UV'$$
$$M = UDV'.$$

Because a limit of orthogonal matrices is an orthogonal matrix and a limit of nonnegative diagonal matrices is a nonnegative diagonal matrix, UDV' is the s.v.d. of M. This completes the proof. □

For each T in $\mathcal{M}$ let

$$\bar{h}(T) = \bar{g}(\nabla f(T) + \alpha T).$$

Then $\bar{h}$ is a point to set mapping on $\mathcal{M}$. Since ∇f is continuous on $\mathcal{M}$, $\nabla f(T) + \alpha T$ is a continuous function of T for T in $\mathcal{M}$. It follows from Luenberger's Corollary 2 (1984, p. 187) that $\bar{h}$ is closed on $\mathcal{M}$. We are ready to prove our basic convergence theorem.

Theorem 3. Let T_n be a sequence of orthogonal matrices in $\mathcal{M}$ such that

$$T_{n+1} \in \bar{h}(T_n).$$

If α is sufficiently large to satisfy condition (3), any limit point of T_n is a stationary point of f restricted to $\mathcal{M}$.

Proof. Using the Zangwill global convergence theorem as given by Luenberger (1984, p. 187), it is sufficient to show that for all $\tilde{T} \in \bar{h}(T)$

$$f(T) < f(\tilde{T}),$$

whenever T is not a stationary point of f restricted to $\mathcal{M}$. This assertion follows from Theorem 1 and completes the proof. □

Now let T_n be the sequence of orthogonal matrices generated by the BSV algorithm. Then $T_{n+1} \in \bar{h}(T_n)$. It follows from Theorem 3 that for α sufficiently large, any limit point of T_n is a stationary point of f restricted to $\mathcal{M}$. This is the global convergence we set out to prove.

We mentioned some modifications of the BSV algorithm in section 3 that used alternatives to the matrix $G + \alpha T$ in step (b). For example, α may be a function of T or αT may be replaced by TW where W is a positive definite matrix that may also be a function of T. If the argument of $\bar{g}$ is modified accordingly in the definition of $\bar{h}$ and it is assumed that α and W are continuous functions of T, Theorem 3 continues to hold.

References

Buck, R.C. (1956). *Advanced calculus*. New York, NY: McGraw-Hill.

Carroll, J.B. (1953). An analytical solution for approximating simple structure in factor analysis. *Psychometrika, 18*, 32–38.

Cliff, N. (1966). Orthogonal rotation to congruence. *Psychometrika, 31*, 33–42.

de Leeuw, J. (1994). Block-relaxation algorithms in statistics. In H.H. Bock, W. Lenski, & M.M. Richter (Eds.), *Information systems and data analysis* (pp. 308–324). Berlin: Springer.

de Leeuw, J., & Pruzansky, S. (1978). A new computational method to fit the weighted Euclidean distance model. *Psychometrika, 43*, 479–490.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*, 1–38.

Golub, G.H., & van Loan, C.F. (1989). *Matrix computations*. London: The Johns Hopkins University Press.

Harman, H.H. (1960). *Modern factor analysis*. Chicago, IL: University of Chicago Press.

Heiser, W.J. (1995). Convergent computation by iterative majorization: Theory and applications in multidimensional data analysis. In W.J. Krzanowski (Ed.), *Recent advances in descriptive multivariate analysis* (pp. 157–189). Oxford: Oxford University Press.

Horst, P. (1965). *Factor analysis of data matrices*. New York, NY: Holt, Rinehart and Winston.

Jamshidian, M., & Jennrich, R.I. (1997). Acceleration of the EM algorithm by using quasi-Newton methods. *Journal of the Royal Statistical Society, Series B, 59*, 569–587.

Kaiser, H.F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika, 23*, 187–200.

Kaplan, W. (1999). *Maxima and minima with applications*. New York, NY: Wiley.

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

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., Takane, Y., & de Leeuw, J. (1990). A generalization of Takane's algorithm for DEDICOM. *Psychometrika, 55*, 151–158.

Loomis, L.H. & Sternberg, S. (1968). *Advanced calculus*. Reading, PA: Addison-Wesley.

Luenberger, D.G. (1984). *Linear and nonlinear programming*. Reading, PA: Addison-Wesley.

Magnus, J.R., & Neudecker, H. (1999). *Matrix differential calculus with applications to statistics and econometrics*. New York, NY: Wiley.

Mulaik, S.A. (1972). *The foundations of factor analysis*. New York, NY: McGraw-Hill.

Neuhaus, J.O., & Wrigley, C. (1954). The quartimax method: An analytical approach to orthogonal simple structure. *British Journal of Statistical Psychology, 7*, 81–91.

Sherin, R.J. (1966). A matrix formulation of Kaiser's varimax criterion. *Psychometrika, 34*, 535–538.

Takane, Y. (1985). Diagonal estimation in DEDICOM. *Proceedings of the 1985 Annual Meeting of the Behaviormetric Society, 13*, 100–101.

ten Berge, J.M.F. (1984). A joint treatment of varimax rotation and the problem of diagonalizing symmetric matrices simultaneously in the least squares sense. *Psychometrika, 49*, 347–358.

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 the singular value approach to varimax rotation. *Computational Statistics Quarterly, 4*, 207–217.

Manuscript received 1 MAR 1999
Final version received 24 APR 2000