Noname manuscript No.
(will be inserted by the editor)

An acceleration method for ten Berge et al.'s algorithm for orthogonal INDSCAL

Yoshio Takane · Kwanghee Jung · Heungsun Hwang

Received: date / Accepted: date

Abstract INDSCAL (INdividual Differences SCALing) is a useful technique for investigating both common and unique aspects of K similarity data matrices. The model postulates a common stimulus configuration in a low-dimensional Euclidean space, while representing differences among the K data matrices by differential weighting of dimensions by different data sources. Since Carroll and Chang proposed their algorithm for INDSCAL, several issues have been raised: non-symmetric solutions, negative saliency weights, and the degeneracy problem. Orthogonal INDSCAL (O-INDSCAL) which imposes orthogonality constraints on the matrix of stimulus configuration has been proposed to overcome some of these difficulties. Two algorithms have been proposed for O-INDSCAL, one by ten Berge, Knol, and Kiers, and the other by Trendafilov. In this paper, an acceleration technique called minimal polynomial extrapolation (MPE) is incorporated in ten Berge et al.'s algorithm. Simulation studies are conducted to compare the performance of the three algorithms (ten Berge et al.'s

The work reported in this paper has been supported by SSHRC Research Grant 36952 to the first author, and by NSERC Discovery Grant 290439 to the third author.

Yoshio Takane
Dept. of Psychology, McGill University, 1205 Dr. Penfield Ave., Montreal, QC H3A 1B1 CANADA
Tel.: 514-398-6125
Fax: 514-398-4896
E-mail: takane@psych.mcgill.ca

Kwanghee Jung
Dept. of Psychology, McGill University, 1205 Dr. Penfield Ave., Montreal, QC H3A 1B1 CANADA
Tel.: 514-398-6135
Fax: 514-398-4896
E-mail: kwanghee.jung@mail.mcgill.ca

Heungsun Hwang
Dept. of Psychology, McGill University, 1205 Dr. Penfield Ave., Montreal, QC H3A 1B1 CANADA
Tel.: 514-398-8021
Fax: 514-398-4896
E-mail: heungsun.hwang@mcgill.ca

original algorithm, the accelerated algorithm, and Trendafilov's). Possible extensions of the accelerated algorithm to similar situations are also suggested.

Keywords multi-way data analysis · minimal polynomial extrapolation (MPE) · singular value decomposition (SVD) algorithm · dynamical system algorithm

PACS 02.70.Rr · 02.70.Uu

Mathematics Subject Classification (2000) 15A15MSC code1 · 15A21 · 15A23

1 Introduction

INDSCAL (INdividual Differences SCALing), proposed by Carroll and Chang (1970), is a useful data analytic tool for investigating the relationships among K symmetric similarity matrices. The model postulates a common stimulus configuration in a low dimensional Euclidean space. The dimensions in the space are then assumed differentially weighted by different "individuals" to give rise to differences among the K matrices. Unlike the simple Euclidean model often used in multidimensional scaling (MDS), the INDSCAL model determines the orientation of coordinate axes uniquely, allowing easy interpretations of the derived stimulus configuration. INDSCAL has been widely used in many areas of science including psychology, sociology, biology, chemistry, etc. See Arabie, Carroll, and DeSarbo (1987) for a practical introduction of INDSCAL. Takane (2007) reviewed some of the most interesting applications of INDSCAL in psychometrics.

Let $\mathbf{S}_k$ (n by n, and $k = 1, \cdots, K$) denote the similarity matrix among n objects obtained from the k^{th} individual. Let $\mathbf{X}$ denote an n by p matrix of stimulus coordinates, where p is the dimensionality of the representation space, and let $\mathbf{D}_k$ (of order p) denote an *nnd* (nonnegative definite) diagonal matrix of saliency weights attached to the p dimensions by individual k. Then the INDSCAL model can be expressed as

$$\mathbf{S}_k = \mathbf{X}\mathbf{D}_k\mathbf{X}' + \mathbf{E}_k, \tag{1}$$

for $k = 1, \cdots K$, where $\mathbf{E}_k$ is the matrix of disturbance terms. To remove the scale indeterminacy between $\mathbf{X}$ and $\mathbf{D}_k$, it is often assumed that

$$\text{diag}(\mathbf{X}'\mathbf{X}) = \mathbf{I}_p, \tag{2}$$

but this is merely a convention. Carroll and Chang (1970) proposed an algorithm to fit the above model based on the alternating least squares (ALS) principle. In an ALS algorithm, the entire set of parameters is divided into several subsets, each of which is updated in turn conditionally on the estimates of other parameters. This cyclic updating process is repeated until convergence is reached. In Carroll and Chang's algorithm, the two $\mathbf{X}$'s in the model are treated as distinct parameters. This is required to ensure monotonic convergence of the algorithm.

Since Carroll and Chang (1970) proposed their algorithm, several issues have been raised. The first issue relates to possible non-symmetric solutions. As noted above, their algorithm treats the two $\mathbf{X}$'s in the model as two distinct sets of parameters. Carroll and Chang assumed that the two estimates of $\mathbf{X}$ were identical at the convergence point due to the symmetry of $\mathbf{S}_k$. However, the identity of the two $\mathbf{X}$'s is by no means guaranteed (ten Berge and Kiers 1991; ten Berge, Kiers, and de Leeuw, 1988). Although in practice, cases in which two estimates of $\mathbf{X}$ differ rarely occur, it is difficult

to prescribe the conditions under which their identity is ensured. Furthermore, the problem is exacerbated when an attempt is made to impose nonnegativity constraints on the dimensional saliency weights in Carroll and Chang's algorithm.

The second issue pertains to possible negative estimates of the saliency weights. Nonnegativity of the saliency weights is important in many applications of INDSCAL in terms of interpretability of solutions. However, there is no mechanism in Carroll and Chang's algorithm to constrain the estimates to be nonnegative. Ten Berge, Kiers, and Krijnen (1993) have shown that negative weights can occur, even when the data matrices are all *nnd* (nonnegative definite). To ensure the nonnegativity of the saliency weights, ten Berge et al. (1993) used the nonnegative least squares method for updating the diagonal elements of $\mathbf{D}_k$. However, this has resulted in frequent occurrences of non-symmetric solutions in which the two estimates of $\mathbf{X}$ differ from each other.

The third issue concerns possible degenerate solutions in INDSCAL. Stegeman (2007) has pointed out that in certain cases the LS criterion to be minimized in INDSCAL does not have a minimum, but only an infimum, within the parameter space. In such situations, no optimal solution exists (Krijnen, Dijkstra, and Stegeman, 2008), and Carroll and Chang's algorithm (or any other algorithm for INDSCAL) is bound to produce a sequence of updates of parameters which are degenerate in the sense that some columns of $\mathbf{X}$ are extremely highly correlated, and some of the diagonal elements of $\mathbf{D}_k$ are arbitrarily large. These "solutions" are totally uninteresting from a substantive viewpoint.

Several attempts have been made to overcome the difficulties. Among them, one promising approach is to modify the original INDSCAL model (1). Specifically, it has been proposed (Kroonenberg, 1983; see also Flury (1988), and Kiers (1989a, 1991)) that the orthogonality constraints,

$$\mathbf{X}'\mathbf{X} = \mathbf{I}_p, \tag{3}$$

be imposed on $\mathbf{X}$ instead of (2). This model is called orthogonal INDSCAL (or O-INDSCAL for short). Krijnen et al. (2008, Theorem 2) have shown that the orthogonality constraints are sufficient to avoid the degeneracy problem mentioned above. It also speeds up the convergence of computational algorithms. (Some evidence to this effect is provided in the empirical result section.) Furthermore, original INDSCAL and O-INDSCAL in most cases give very similar results. In order to get a glimpse of it, six published data sets were analyzed by the two models. The results are summarized in Table 1. It can be seen that in all cases solutions were virtually identical. Correlations between corresponding dimensions averaged over all dimensions were extremely high (at least .991) for both stimulus coordinates and saliency weights. The last column of the table shows the absolute correlations between the stimulus coordinates obtained by original INDSCAL. (In the case of three-dimensional solutions, the mean of the absolute correlations were calculated over three pairs of dimensions.) They were all very small (the largest is .111), indicating that original INDSCAL in most cases yields nearly orthogonal solutions without explicitly requiring the orthogonality. This implies that imposing the orthogonality constraints tends to be fairly innocuous.

Two algorithms have been developed for O-INDSCAL (ten Berge, Knol, and Kiers, 1988; Trendafilov, 2004). Both of these algorithms are guaranteed to obtain symmetric solutions, and are capable of imposing nonnegativity restrictions on the saliency weights. In this paper, an acceleration technique called minimal polynomial extrapolation (MPE; Smith, Ford, and Sidi, 1987) is incorporated in ten Berge et al.'s (1988)

Table 1 Average correlations between corresponding dimensions in original INDSCAL and O-INDSCAL, and (average) absolute correlations between dimensions in original INDSCAL.

source	stimuli	K	n	p	(mean) corr. coord.	(mean) corr. weights	(mean) abs. corr. bet. dim.
Helm (1964)	colors	16	10	2	.9926	.9989	.1110
Jacobowitz (1975)	body parts	30	15	3	.9910	.9961	.0990
Rosenberg et al. (1975)	kinships	6	15	3	.9937	.9998	.0190
Schiffman et al. (1981)	colas	10	10	3	.9994	.9910	.0513
Nguyen et al. (2008)	"blicks"	18	8	2	.9999	1.0000	.0230
Zhang et al. (in press)	sports	10	8	2	.9990	.9998	.0490

algorithm to further speed up its convergence. It will be shown through simulation studies that the accelerated algorithm is more efficient than ten Berge et al.'s original algorithm as well as Trendafilov's (2004). These algorithms are briefly reviewed in the next section, followed by a description of the acceleration technique incorporated into ten Berge et al.'s algorithm. The performance of these algorithms is compared in subsequent sections. The final section discusses possible extensions of the accelerated algorithm to similar situations.

2 Existing Algorithms

In this section, two existing algorithms for O-INDSCAL are briefly reviewed, one by ten Berge, Knol, and Kiers (1988), and the other by Trendafilov (2004).

2.1 Ten Berge et al.'s (1988) algorithm

Parameters in the O-INDSCAL model are estimated in such a way that the least squares (LS) criterion,

$$\phi(\mathbf{X}, \mathbf{W}) = \sum_{k=1}^{K} \mathrm{SS}(\mathbf{E}_k) = \sum_{k=1}^{K} \mathrm{tr}(\mathbf{E}_k' \mathbf{E}_k), \tag{4}$$

is minimized subject to the orthogonalization restriction (3), where $\mathbf{E}_k = \mathbf{S}_k - \mathbf{X}\mathbf{D}_k\mathbf{X}'$, and $\mathbf{W}$ is a p by K matrix of saliency weights with diagonal elements of $\mathbf{D}_k$ as the k^{th} column vector. The minimization is done by first minimizing (4) with respect to $\mathbf{D}_k$ conditional on $\mathbf{X}$, and then with respect to $\mathbf{X}$. That is,

$$\min_{X,W} \phi(\mathbf{X}, \mathbf{W}) = \min_{X} \min_{W|X} \phi(\mathbf{X}, \mathbf{W}). \tag{5}$$

Let $\phi_k(\mathbf{X}, \mathbf{D}_k) = \mathrm{SS}(\mathbf{S}_k - \mathbf{X}\mathbf{D}_k\mathbf{X}')$. Then (4) can be rewritten as

$$\phi(\mathbf{X}, \mathbf{W}) = \sum_{k=1}^{K} \phi_k(\mathbf{X}, \mathbf{D}_k), \tag{6}$$

and

$$\min_{W|X} \phi(\mathbf{X}, \mathbf{W}) = \sum_{k=1}^{K} \min_{D_k|X} \phi_k(\mathbf{X}, \mathbf{D}_k). \tag{7}$$

The conditional minimum of $\phi_k(\mathbf{X}, \mathbf{D}_k)$ with respect to $\mathbf{D}_k$ given $\mathbf{X}$ and subject to the nonnegativity restrictions is obtained by

$$\hat{\mathbf{D}}_k = \max\{\mathbf{0}, \operatorname{diag}(\mathbf{X}'\mathbf{S}_k\mathbf{X})\}. \tag{8}$$

This means that a diagonal element of $\hat{\mathbf{D}}_k$ is equal to the corresponding element of $\operatorname{diag}(\mathbf{X}'\mathbf{S}_k\mathbf{X})$ if the latter is nonnegative. However, it is set equal to 0 if the latter is negative. This simplified nonnegative LS procedure is justified by the fact that the LS criterion is separable with respect to the diagonal elements of $\mathbf{D}_k$ given $\mathbf{X}$.

Let

$$\phi^*(\mathbf{X}) = \sum_{k=1}^{K} \min_{D_k|X} \phi_k(\mathbf{X}, \mathbf{D}_k) = \sum_{k=1}^{K} \phi_k(\mathbf{X}, \hat{\mathbf{D}}_k), \tag{9}$$

where

$$\phi_k(\mathbf{X}, \hat{\mathbf{D}}_k) = \min_{D_k|X} \phi_k(\mathbf{X}, \mathbf{D}_k). \tag{10}$$

To obtain the minimum of $\phi^*(\mathbf{X})$ with respect to $\mathbf{X}$ subject to the orthogonality constraint (3), we use the Lagrange multiplier method. Let

$$\phi^{**}(\mathbf{X}) = \phi^*(\mathbf{X}) + 2\operatorname{tr}(\mathbf{S}(\mathbf{X}'\mathbf{X} - \mathbf{I})), \tag{11}$$

where $\mathbf{S}$ is a symmetric matrix of Lagrange multipliers. By differentiating $\phi^{**}(\mathbf{X})$ with respect to $\mathbf{X}$, and setting the results equal to $\mathbf{0}$, we obtain

$$-\frac{1}{4}\frac{\partial \phi^{**}(\mathbf{X})}{\partial \mathbf{X}} = \mathbf{G} - \mathbf{X}(\sum_{k=1}^{K} \hat{\mathbf{D}}_k^2) - \mathbf{X}\mathbf{S} = \mathbf{0}, \tag{12}$$

where

$$\mathbf{G} = \sum_{k=1}^{K} \mathbf{S}_k\mathbf{X}\hat{\mathbf{D}}_k. \tag{13}$$

Note that in deriving (12), $\hat{\mathbf{D}}_k$ is treated as if it were a constant matrix, while it is in fact a function of $\mathbf{X}$. This is justified by the fact that $\hat{\mathbf{D}}_k$ has been obtained in such a way as to minimize $\phi_k(\mathbf{X}, \mathbf{D}_k)$ conditional on $\mathbf{X}$[1]. (Details of the justification are given in the Appendix.) By premultiplying both sides of (12) by $\mathbf{X}'$, we obtain

$$\mathbf{S} = \mathbf{X}'\mathbf{G} - \sum_{k=1}^{K} \hat{\mathbf{D}}_k^2, \tag{14}$$

[1] The derivation of ten Berge et al.'s (1988) algorithm presented here is slightly more general than their original derivation. They assumed that the conditional minimum of $\mathbf{D}_k$ given $\mathbf{X}$ was given by $\hat{\mathbf{D}}_k = \operatorname{diag}(\mathbf{X}'\mathbf{S}_k\mathbf{X})$. However, strictly speaking, this does not allow us to impose nonnegativity restrictions on $\hat{\mathbf{D}}_k$. Our formulation, on the other hand, works no matter what $\hat{\mathbf{D}}_k$ is, as long as it satisfies $\min_{D_k|X} \phi_k(\mathbf{X}, \mathbf{D}_k)$. Even a closed form expression for $\hat{\mathbf{D}}_k$ is unnecessary.

and by putting this expression of $\mathbf{S}$ into (12), we obtain

$$(\mathbf{I} - \mathbf{X}\mathbf{X}')\mathbf{G} = \mathbf{0} \tag{15}$$

as the stationary equation to be solved. This equation is solved by (e.g., Jennrich, 2001)

$$\mathbf{X} = \mathbf{G}(\mathbf{G}'\mathbf{G})^{-1/2} = \mathbf{U}\mathbf{V}', \tag{16}$$

where $\mathbf{U}$ and $\mathbf{V}$ are such that $\mathbf{G} = \mathbf{U}\Delta\mathbf{V}'$ is the SVD of $\mathbf{G}$. (It can be readily verified that $\mathbf{X}$ given above satisfies (15), and that $\mathbf{X}'\mathbf{G} = \mathbf{V}\mathbf{U}'\mathbf{U}\Delta\mathbf{V}' = \mathbf{V}\Delta\mathbf{V}'$ is indeed symmetric.) Matrix $\mathbf{G}$ is iteratively updated by (13), which is orthogonalized by (16) to obtain a new $\mathbf{X}$. This iterative updating scheme is repeated until the Frobenius norm of (15) is smaller than a certain prescribed value.

2.2 Trendafilov's algorithm

Trendafilov (2004) proposed an algorithm for O-INDSCAL with nonnegativity restrictions on the saliency weights. The algorithm minimizes the same criterion (4) subject to the same orthogonality and nonnegativity restrictions on $\mathbf{D}_k$ as before. Partial derivatives of (4) with respect to model parameters $\mathbf{X}$ and $\mathbf{D}_k$ are taken, which are set equal to zero. The resultant nonlinear equations are then solved iteratively. Contrary to ten Berge et al., however, Trendafilov does not follow the conditional minimization strategy. Instead, the criterion is minimized simultaneously with respect to both $\mathbf{X}$ and $\mathbf{D}_k$.

To incorporate the nonnegativity restrictions on $\mathbf{D}_k$, $\mathbf{D}_k$ in the model is reparameterized as $\tilde{\mathbf{D}}_k^2$. The idea is to estimate $\tilde{\mathbf{D}}_k$ directly. Then $\mathbf{D}_k = \tilde{\mathbf{D}}_k^2$ is always nonnegative without explicitly imposing $\mathbf{D}_k \geq \mathbf{0}$. Partial derivatives of (4) with respect to $\mathbf{X}$ are given by

$$-\frac{1}{4}\frac{\partial \phi}{\partial \mathbf{X}} = \mathbf{G} - \mathbf{X}(\sum_{k=1}^{K} \tilde{\mathbf{D}}_k^4), \tag{17}$$

where

$$\mathbf{G} = \sum_{k=1}^{K} \mathbf{S}_k \mathbf{X} \tilde{\mathbf{D}}_k^2. \tag{18}$$

(Equation (17) is essentially the same as the first two terms in (12), except that $\hat{\mathbf{D}}_k^2$ in the former is replaced by $\tilde{\mathbf{D}}_k^4$ in the latter.) To incorporate the orthogonality constraints on $\mathbf{X}$, the above partial derivatives are projected onto the space tangent to the Stiefel manifold at $\mathbf{X}$. The projected (negative) gradients are given by

$$-\mathbf{g}(\mathbf{X}) = \mathbf{G} - \mathbf{X}(\mathbf{X}'\mathbf{G} + \mathbf{G}'\mathbf{X})/2 \tag{19}$$

(Trendafilov, 2004; see also Jennrich (2001)). (In ten Berge et al.'s (1988) algorithm, $-\mathbf{g}(\mathbf{X})$ is set equal to $\mathbf{0}$, and the resultant equation is solved for $\mathbf{X}$. Since $\mathbf{X}'\mathbf{G}$ is symmetric at $-\mathbf{g}(\mathbf{X}) = \mathbf{0}$, the latter reduces to (15), which indicates a close relationship between the two algorithms.) Partial derivatives of (4) with respect to $\tilde{\mathbf{D}}_k$, on the other hand, are given by

$$-\mathbf{g}(\tilde{\mathbf{D}}_k) = -\frac{1}{4}\frac{\partial \phi}{\partial \tilde{\mathbf{D}}_k} = (\mathbf{X}'\mathbf{S}_k\mathbf{X}) \odot \tilde{\mathbf{D}}_k - \tilde{\mathbf{D}}_k^3, \tag{20}$$

where $\odot$ indicates elementwise multiplications.

These gradients may be used to define a steepest descent step to minimize (4). Trendafilov (2004), however, used its continuous analog, called the dynamical system algorithm, in which instantaneous changes in $\mathbf{X}$ and $\tilde{\mathbf{D}}_k$ over time are equated with the negative gradients given above, i.e.,

$$\frac{d\mathbf{X}}{dt} = -\mathbf{g}(\mathbf{X}), \quad \text{and} \quad \frac{d\tilde{\mathbf{D}}_k}{dt} = -\mathbf{g}(\tilde{\mathbf{D}}_k). \tag{21}$$

These differential equations define a continuous steepest descent flow of parameter updates toward a local minimum of the LS loss function (4). The algorithm essentially solves the ordinary differential equations (ODE) specified by (21) starting from some initial values. In Trendafilov's computer program, a numerical integration routine is repeatedly applied in the time interval of $[0, 249]$ in step of $\Delta t = 25$ until an improvement between two consecutive model fits is less than a certain small number.

3 The Minimal Polynomial Extrapolation (MPE) Method

In this section we describe the minimal polynomial extrapolation (MPE) method (Smith, Ford, and Sidi, 1987; see also Loisel and Takane (2009), and Takane and Zhang (in press)), and how it is incorporated into ten Berge et al.'s algorithm to accelerate its convergence. In our exposition, we emphasize the interpretation of the MPE method as a kind of preconditioner (e.g., Golub and van Loan, 1996). Matrix $\mathbf{B}$ is called a preconditioner, when for solving $\mathbf{Ax} = \mathbf{d}$, $\mathbf{BAx} = \mathbf{Bd}$ is solved instead, where $\mathbf{BA}$ has a much smaller condition number than $\mathbf{A}$. A somewhat broader perspective of preconditioning is that both sides of equation $\mathbf{BAx} = \mathbf{Bd}$ is simpler in some sense than those in the original equation. As will be shown, the MPE method can be regarded as such a procedure.

In the MPE acceleration method, a sequence of updates of parameters is examined, and the convergence point of the sequence is predicted as accurately as possible. For illustration, the updating equation is temporarily assumed linear. Let

$$\mathbf{x}^{(q+1)} = \mathbf{H}\mathbf{x}^{(q)} + \mathbf{b} \tag{22}$$

denote the updating equation, where the superscript q indicates an iteration number. It is assumed that the largest absolute eigenvalue of $\mathbf{H}$ is strictly less than unity for the above iterative process to converge to some stationary point of $\mathbf{x}$. (Throughout this exposition, a vectorized form of parameters is used for convenience.) Let $\mathbf{x}_{conv}$ denote a stationary point of (22). Then,

$$(\mathbf{I} - \mathbf{H})\mathbf{x}_{conv} = \mathbf{b}. \tag{23}$$

This equation can be solved by

$$\mathbf{x}_{conv} = (\mathbf{I} - \mathbf{H})^{-1}\mathbf{b}, \tag{24}$$

where $\mathbf{I} - \mathbf{H}$ is assumed nonsingular, provided that $\mathbf{H}$ can be explicitly calculated, and its order is not exceedingly large. However, in some cases, either one or both of these conditions may fail (e.g., $\mathbf{H}$ exists only in algorithmic form, only a product of the form $\mathbf{Hx}$ is computable for some vector $\mathbf{x}$, or the size of $\mathbf{H}$ is prohibitively large for even an implicit inversion of $\mathbf{I} - \mathbf{H}$), and an iterative scheme like (22) may have to be used.

In the MPE method, equation (23) is solved without calculating the inverse of $\mathbf{I}-\mathbf{H}$ (even implicitly).

Let $\mathbf{u}^{(q)} = \mathbf{x}^{(q+1)} - \mathbf{x}^{(q)}$ $(q = 0, \cdots, j)$, where the first j vectors are assumed linearly independent, whereas the entire $j+1$ vectors are linearly dependent. That is, there exist c_q $(q = 0, \cdots, j)$ such that

$$\sum_{q=0}^{j} c_q \mathbf{u}^{(q)} = \mathbf{0}, \tag{25}$$

where it is further assumed that $c_j = 1$ (without loss of generality). Such a j always exists for any set of a sufficiently large number of vectors. For example, for a set of m-component vectors, $j \leq m$ always holds. Let $\mathbf{c} = [c_0, c_1, \cdots, c_{j-1}]'$, $\mathbf{U} = [\mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \cdots, \mathbf{u}^{(j-1)}]$. Then

$$\mathbf{c} = -\mathbf{U}^{+}\mathbf{u}^{(j)}, \tag{26}$$

where $\mathbf{U}^{+}$ is the Moore-Penrose inverse of $\mathbf{U}$. (The $\mathbf{c}$ obtained above is used later in (31) and (32) to derive $\mathbf{x}_{conv}$.)

Using the relation that $\mathbf{u}^{(q)} = \mathbf{H}^q\mathbf{u}^{(0)}$, (25) can be rewritten in the form of a polynomial equation in $\mathbf{H}$. That is,

$$\mathbf{0} = \sum_{q=0}^{j} c_q \mathbf{u}^{(q)} = \sum_{q=0}^{j} c_q \mathbf{H}^q \mathbf{u}^{(0)} = \mathbf{P}(\mathbf{H})\mathbf{u}^{(0)}. \tag{27}$$

Since $\mathbf{u}^{(0)}$ is in the null space of $\mathbf{P}(\mathbf{H})$, $\mathbf{P}(\mathbf{H})$ is said to be the minimal polynomial function of $\mathbf{H}$ that annihilates $\mathbf{u}^{(0)}$.

Notice that (23) can be rewritten as

$$(\mathbf{I}-\mathbf{H})\mathbf{x}_{conv} = (\mathbf{I}-\mathbf{H})\mathbf{x}^{(0)} + \mathbf{u}^{(0)}. \tag{28}$$

(This identity can be readily verified by noting that the righthand side is equal to $\mathbf{x}^{(0)} - \mathbf{H}\mathbf{x}^{(0)} + \mathbf{u}^{(0)} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)} + \mathbf{b} + \mathbf{x}^{(1)} - \mathbf{x}^{(0)} = \mathbf{b}$.) By premultiplying both sides of (28) by $\mathbf{P}(\mathbf{H})$, we obtain

$$\mathbf{P}(\mathbf{H})(\mathbf{I}-\mathbf{H})\mathbf{x}_{conv} = \mathbf{P}(\mathbf{H})(\mathbf{I}-\mathbf{H})\mathbf{x}^{(0)} \tag{29}$$

because $\mathbf{P}(\mathbf{H})\mathbf{u}^{(0)} = \mathbf{0}$ by (27). Note that $\mathbf{I}-\mathbf{H}$ commutes with $\mathbf{P}(\mathbf{H})$, and is also invertible as has been assumed earlier. This allows us to eliminate $\mathbf{I}-\mathbf{H}$ entirely from the above equation. By premultiplying both sides of (29) by $(\mathbf{I}-\mathbf{H})^{-1}$, we obtain

$$\mathbf{P}(\mathbf{H})\mathbf{x}_{conv} = \mathbf{P}(\mathbf{H})\mathbf{x}^{(0)}. \tag{30}$$

Since $\mathbf{H}\mathbf{x}_{conv} = \mathbf{x}_{conv} - \mathbf{b}$, and $\mathbf{H}\mathbf{x}^{(0)} = \mathbf{x}^{(1)} - \mathbf{b}$ (or more generally $\mathbf{H}\mathbf{x}^{(q)} = \mathbf{x}^{(q+1)} - \mathbf{b}$ for $q = 0, \cdots, j$), equation (30) can be further simplified into

$$(\sum_{q=0}^{j} c_q)\mathbf{x}_{conv} = \sum_{q=0}^{j} c_q \mathbf{x}^{(q)}, \tag{31}$$

leading to

$$\mathbf{x}_{conv} = \sum_{q=0}^{j} c_q \mathbf{x}^{(q)} / \sum_{q=0}^{j} c_q. \tag{32}$$

as the solution of (28) and of (23). Equations (29) and (30) show that $\mathbf{P}(\mathbf{H})$ acts like a preconditioner on $\mathbf{I}-\mathbf{H}$ that simplifies the solution of the original equation (23). (That is, $\mathbf{P}(\mathbf{H})$ annihilates $\mathbf{u}^{(0)}$ in the right hand side of (28), which allows to eliminate $\mathbf{I}-\mathbf{H}$ from both sides of equation (29) to obtain equation (30), which is further simplified into (31).)

The steps in the MPE acceleration method are summarized as follows for the linear case:

Step (1) Initialize $\mathbf{x}$ by $\mathbf{x}^{(0)}$.
Step (2) Generate $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(j+1)}$ according to ten Berge et al.'s (1988) procedure described in section 2.1.
Step (3) Calculate $c_0, \cdots, c_{j-1}, c_j(=1)$ based on the sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(j+1)}$ according to (26).
Step (4) Obtain $\mathbf{x}_{conv}$ by (32) using c_j's calculated in Step (3).

The above steps give the exact solution of $\mathbf{x}$, provided that the basic iterate is linear as has been assumed, and that the exact value of j is known. However, since ten Berge et al.'s (1988) algorithm involves nonlinear updating equations, the above steps have to be applied repeatedly. The $\mathbf{x}_{conv}$ obtained in Step (4) is re-orthogonalized by SVD and used as $\mathbf{x}^{(0)}$ in the next cycle of iterations. The exact value of j is also usually unknown in advance. Fortunately, its exact value is not so crucial. It can be any value for which the relation (25) holds approximately, as far as the above procedure has to be applied repeatedly because of the nonlinearity of the updating equations. Typically, there is a wide range of values of j for which the MPE method works reasonably well. For finding the best j, several values of j (= 5, 10, and 15) are tried in the following empirical studies.

Unfortunately, there is not very much one can say about the convergence properties of the MPE method in nonlinear cases, and when j is used for which (25) is only approximately true. Some attempts have been made to develop a theory for these difficult cases. For example, Smith et al. (1987, sections 6 and 7) have shown that under some conditions the MPE update has a quadratic convergence rate. However, these conditions are difficult to verify in practice. Nonetheless, our experience with the algorithm indicates that the repetitive applications of the above steps work well in most cases, indicating that our basic iterate is nearly linear at least locally in a neighborhood of a convergence point.

4 Numerical Experiments

In this section, the results of simulation studies are reported that demonstrate 1) the computational advantage of O-INDSCAL over the original INDSCAL, and 2) the efficiency of the accelerated algorithm (hereafter called the MPE algorithm) relative to the non-accelerated algorithms. In the first study, ten Berge et al.'s (1988) algorithm for O-INDSCAL is compared with an algorithm for original INDSCAL called SYMPRES (ten Berge, Knol, and Kiers, 1993), which is able to obtain symmetric solutions under nonnegativity restrictions on saliency weights. In the second study, the MPE algorithm is compared with ten Berge et al.'s (1988) algorithm, and Trendafilov's algorithm for O-INDSCAL. A limited study investigating the seriousness of suboptimal solutions in O-INDSCAL has also been conducted (Study 3).

In generating data sets for these studies, the number of matrices (K) and the number of variables (n) were each manipulated at two levels: $K = 10$ and 30, and $n = 10$ and 50, which were factorially combined to generate three different kinds of data sets. One was completely random, and the other two were partially structured. The completely random data sets were generated according to the standard normal distribution for each entry of $\mathbf{S}_k$, followed by a symmetrizing operation (i.e., $(\mathbf{S}_k + \mathbf{S}_k')/2$). In the structured data sets, $\mathbf{X}$ and $\mathbf{D}_k$ were first randomly generated, from which $\mathbf{S}_k$ ($k = 1, \cdots, K$) were calculated according to (1). The number of components were varied at two levels: $p = 3$ and 5. Elements of $\mathbf{X}$ were generated by uniform random numbers between 0 and 1, which were then orthogonalized by SVD. Diagonal elements of $\mathbf{D}_k$ were generated according to the standard normal distribution. In one kind of structured data sets, negative weights were left as they were (the indefinite case), while in the other they were turned into positive weights by taking the absolute value (the nonnegative definite (*nnd*) case). Elements of $\mathbf{E}_k$ were assumed to follow a normal distribution with 0 mean and a variance of σ^2. The value of σ was set to 10% of the standard deviation of the structural part ($\mathbf{X}\mathbf{D}_k\mathbf{X}'$) of the model.

4.1 Time comparisons: Studies 1 and 2

The maximum number of iterations was set to 1000 for the completely random data sets and 500 for both kinds (*nnd* and indefinite weight matrices) of structured data sets. All analyses were carried out under the assumptions of $p = 3$ and $p = 5$, which were the correct dimensionalities (the number of components) for the structured data sets, but were not "correct" for the completely random data sets. To evaluate the efficiency of the algorithms, the mean CPU time was measured based on 100 replicated data sets. All the experiments were implemented in MATLAB 7.0, and the computer we used was an Intel Core 2 Duo CPU E7300 @ 2.66GHz with 2 GB of RAM.

In the first study, ten Berge et al.'s (1988) algorithm for O-INDSCAL used the convergence criterion of $||\mathbf{G}_p|| = \mathrm{tr}(\mathbf{G}_p'\mathbf{G}_p)^{1/2} < 10^{-6}$, where $\mathbf{G}_p = (\mathbf{I} - \mathbf{X}\mathbf{X}')\mathbf{G}$, and $\mathbf{G}$ was as defined in (13), while SYMPRES (ten Berge et al., 1993) for original INDSCAL used $||\mathbf{G} - \mathbf{X}(\sum_{k=1}^{K} \hat{\mathbf{D}}_k)|| < 10^{-6}$. Table 2 reports the mean CPU time for the two algorithms for the three kinds of data sets (completely random (RAND), partially structured with indefinite weight matrices (IND), and partially structured with *nnd* weight matrices (NND)). As can be observed, ten Berge et al.'s algorithm for O-INDSCAL is clearly (roughly 3 to 5 times) faster than SYMPRES, which is considered to be a state-of-the-art algorithm for original INDSCAL.

In the second study, the MPE algorithm was compared with ten Berge et al.'s (1988) algorithm, and Trendafilov's (2004) algorithm for O-INDSCAL. In analyzing the data sets by the MPE algorithm, several values of $j = 5, 10$, and 15 were tried to see which value gave the best performance. For all three algorithms, the initial $\mathbf{X}$ was first generated by uniform random numbers, which was then orthogonalized by SVD. The initial $\mathbf{D}_k$ was then calculated by (8).

Ten Berge et al.'s algorithm and the MPE algorithm used the same convergence criterion that the former algorithm used in the first study. This represents a very stringent convergence criterion. Trendafilov's algorithm, on the other hand, used the criterion that the improvement in the value of the fitting criterion in two successive iterations was less than 10^{-4}. This convergence criterion was found to be much more lenient than the one used in the other algorithms. However, no further effort was made

Table 2 The mean CPU time for ten Berge et al.'s (1993) for original INDSCAL (SYMPRES), and ten Berge et al.'s (1988) original algorithm for O-INDSCAL with three kinds of generated data sets: completely random (RAND), indefinite weight matrices (IND), and nonnegative definite weight matrices (NND).

			RAND		IND		NND	
K	n	p	SYMPRES	ten Berge	SYMPRES	ten Berge	SYMPRES	ten Berge
10	10	3	0.3040	0.0780	0.1831	0.0226	0.0367	0.0126
10	10	5	1.3558	0.1686	0.6599	0.0696	0.0948	0.0335
10	50	3	1.6530	0.6401	1.9174	0.1317	0.3423	0.0864
10	50	5	6.7686	1.0622	4.9616	0.1777	0.5961	0.1221
30	10	3	0.4067	0.1356	0.1509	0.0147	0.0727	0.0431
30	10	5	0.9850	0.1856	0.4432	0.0307	0.1225	0.0577
30	50	3	3.9392	2.6203	0.4355	0.1009	0.4025	0.2763
30	50	5	6.5600	3.2256	1.2416	0.1345	0.6906	0.3752

Table 3 The mean CPU time of ten Berge et al.'s (1988) original algorithm, the MPE algorithm ($j = 5, 10$, and 15), and Trendafilov's algorithm with completely random data sets.

			ten Berge et al.	The MPE algorithm			Trendafilov
K	n	p		$j = 5$	$j = 10$	$j = 15$	
10	10	3	0.0780	0.0270	0.0298	0.0253	0.2430
10	10	5	0.1686	0.0427	0.0424	0.0451	0.4359
10	50	3	0.6401	0.2613	0.2652	0.2757	1.6146
10	50	5	1.0622	0.2764	0.3374	0.3718	6.6845
30	10	3	0.1356	0.0853	0.1120	0.1136	1.0798
30	10	5	0.1856	0.1033	0.1165	0.1356	2.9192
30	50	3	2.6203	1.0803	1.6104	1.3096	8.3058
30	50	5	3.2255	1.4379	1.8792	1.9912	38.6959

to make the two convergence criteria more comparable because the former algorithms were found to be much faster even with the more stringent convergence criterion.

Table 3 shows the mean CPU time required for convergence over 100 replications of the completely random data sets. As can be seen, an optimal value of j for the MPE algorithm is around 5. In this case, the mean CPU time was reduced to 50% to 70% of that of ten Berge's algorithm, compared to ten Berge et al.'s algorithm ($j = 1$), which in turn is much faster than Trendafilov's algorithm. This is despite the fact that a much more lenient convergence criterion is used in the latter.

It should be noted that Trendafilov's algorithm sometimes (about 40% of the time) obtained a value of the fitting criterion slightly smaller than that obtained by ten Berge et al.'s and the MPE algorithms. However, it has been found that this is due to the fact that Trendafilov's algorithm does not ensure strict orthogonality of $\mathbf{X}$ at the convergence point. A smaller fitting value thus does not imply that his algorithm obtains a more precise solution. In fact, when the norm of the projected gradients was evaluated at Trendafilov's solution (giving a smaller value of ϕ), it was invariably found to be larger than those for solutions from ten Berge et al.'s and the MPE algorithms.

Table 4 displays the mean CPU time required over 100 replications of the structured data sets generated with indefinite weight matrices. In all cases, the analysis was carried out under "correct" dimensionalities (i.e., if the data were generated with $p = 3$, for

Table 4 The mean CPU time of ten Berge et al.'s (1988) original algorithm, and the MPE algorithm ($j = 5, 10$, and 15), and Trendafilov's algorithm for structured data sets generated with indefinite weight matrices.

			ten Berge et al.	The MPE algorithm			Trendafilov
K	n	p		$j = 5$	$j = 10$	$j = 15$	
10	10	3	0.0226	0.0069	0.0066	0.0111	0.1853
10	10	5	0.0696	0.0112	0.0125	0.0193	0.3212
10	50	3	0.1317	0.0250	0.0301	0.0404	0.9491
10	50	5	0.1777	0.0303	0.0426	0.0449	3.2197
30	10	3	0.0147	0.0090	0.0098	0.0115	0.6175
30	10	5	0.0307	0.0145	0.0150	0.0164	1.3352
30	50	3	0.1009	0.0590	0.0643	0.0849	2.7494
30	50	5	0.1345	0.0856	0.0850	0.1025	10.4758

Table 5 The mean CPU time of ten Berge et al.'s (1988) original algorithm, the MPE algorithm ($j = 5, 10$, and 15), and Trendafilov's algorithm with structured data sets generated with *nnd* weight matrices.

			ten Berge et al.	The MPE algorithm			Trendafilov
K	n	p		$j = 5$	$j = 10$	$j = 15$	
10	10	3	0.0126	0.0078	0.0073	0.0078	0.0973
10	10	5	0.0335	0.0148	0.0117	0.0126	0.1535
10	50	3	0.0864	0.0304	0.0293	0.0342	0.3725
10	50	5	0.1221	0.0544	0.0495	0.0537	0.9795
30	10	3	0.0431	0.0198	0.0198	0.0212	0.2841
30	10	5	0.0577	0.0259	0.0264	0.0309	0.5504
30	50	3	0.2763	0.1451	0.1318	0.1201	1.1073
30	50	5	0.3752	0.2009	0.1735	0.1966	2.8238

example, they were analyzed under the assumption of $p = 3$). As can be seen, the mean CPU time is overall much shorter than in Table 3. This, of course, is due to the structures built into the data. The table also shows a substantial reduction (50% to 70%) in the mean CPU time by the MPE algorithm, compared with the other algorithms. Again, an optimal value of j for the MPE algorithm is found to be near 5. It should be noted that the CPU time for ten Berge et al.'s algorithm is uniformly larger for $K = 10$ than for $K = 30$, which may look odd. However, this was mainly caused by a few data sets in $K = 10$ for which ten Berge et al.'s algorithm did not converge within the 500 iteration limit.

Finally, Table 5 gives the mean CPU time for the second kind of structured data sets generated with *nnd* weight matrices. Again all analyses were done under the correct dimensionality specifications. The results are very similar to those given in the previous table. The MPE algorithm achieved a substantial reduction in time for convergence, compared with the other algorithms. In the present case, an optimal value of j is found to be around 10, and the $\mathbf{S}_k$ matrices are all nearly *nnd*. However, no appreciable gain was obtained by this in convergence speed.

4.2 Seriousness of suboptimal solutions: Study 3

So far, the mean CPU time has been calculated based on the first stationary point the algorithms have found, but no attention has been paid to the quality of the stationary point, i.e., if it is a global minimum, or other stationary points. (Other stationary points include both mere local minima and saddle points. See Dosse and ten Berge (2008) for the possibility of the latter.) We now systematically investigate the chances of getting a global minimum in a single run, and figure out the minimum number of runs needed to ensure a global minimum at least once with a certain probability (say, with probability at least as large as .999) for each of the three kinds of data sets discussed above (completely random, and two kinds of structured data sets).

For this study, 50 replicated data sets were generated for each of the three kinds of data sets with $K = 30$, and $n = 10$, and for the two kinds of structured data sets, with $p = 3$. Each of these 50 replicated data sets was analyzed by the MPE algorithm under $p = 3$ with 50 different random initials to find globally optimal solutions. The probability of finding a global minimum at least once out of 50 runs is estimated to be quite high. (It is at least $1 - .75^{50} > .999999$, assuming that the probability of not finding a global minimum in a single run is .75, which is indeed the probability of not finding a global minimum in a single run in the most difficult situation that has been tried, i.e., three dimensional solutions for completely random data.) Based on the global minimum solutions thus obtained, the probability of not finding a global minimum in a single run was estimated. This information was then used to calculate the minimum number of runs needed to obtain a global minimum solution at least once with probability .999 or larger.

The rate of suboptimal solutions in a single run by the three algorithms was found to be about the same, and about 75% (75.4% in ten Berge et al.'s original algorithm, 76.7% at $j = 5$, 75.9% at $j = 10$, and 74.8% at $j = 15$, and 74% in Trendafilov's algorithm) for completely random data with $K = 30$, and $n = 10$. These numbers are somewhat daunting. However, it should be reminded that these are for completely random data sets. Assuming the probability of not finding a global minimum is about .75, approximately 25 runs with random initials are necessary to obtain a global minimum at least once with probability $1 - .75^{25} > .999$. (To avoid repetitions, the qualification of the statement by "with probability greater than .999" will be omitted in the sequel.) It is also observed that the acceleration did not help reduce the probability of hitting suboptimal solutions.

The problem of convergence to suboptimal solutions was also investigated with two kinds of structured data sets. With the data sets generated by indefinite weight matrices, the rate of convergence to suboptimal solutions was very low, less than 1% (0.56% in ten Berge et al.'s original algorithm, 0.80% at $j = 5$, 0.44% at $j = 10$, and 0.16% at $j = 15$) when the correct dimensionality of 3 was specified. This requires at most 2 runs for hitting a global minimum at least once. Trendafilov's algorithm, on the other hand, hits a suboptimal solution under the same condition with the probability of approximately .20, which translates into a minimum of 5 runs to ensure a global minimum.

The picture is entirely different, however, when the dimensionality of solutions is misspecified. When the same sets of data as above were analyzed under $p = 2$, the chance of suboptimal solutions went up to 50% to 55% (51.6% in ten Berge et al.'s original algorithm, 53.2% at $j = 5$, 53.2% at $j = 10$, and 52.3% at $j = 15$). This requires about 10 to 12 runs to ensure a global minimum solution. Under the same

condition, Trendafilov's algorithm hits suboptimal solutions at the rate of approximately 45%, which translats into 9 runs to ensure a globally optimal solution. When the dimensionality of solutions was overestimated (i.e., the analysis was done under $p = 4$), all three algorithms hit suboptimal solutions about 40% of the times. This translates into approximately 8 runs. The results with *nnd* weight matrices are very similar to those with indefinite weight matrices except that the chance of suboptimal solutions in Trendafilov's algorithm was as low as ten Berge et al.'s and the MPE algorithms (less than 2%) when the correct dimensionality was specified, and will not be presented separately.

One should be warned that the above investigation is quite limited in scope with $K = 30$, $n = 10$, and $p = 3$ only. The rate of suboptimal solutions depends crucially on these factors, and each specific case should be examined individually. Still, some general tendency has emerged from the above study. One obvious point is that the chance of suboptimal solutions is much lower when an approximately correct model is fitted. This information may be usefully exploited in determining the appropriate number of dimensions in the solution.

5 Discussion

This paper began with a discussion on several issues associated with the original INDSCAL model and algorithm (i.e., non-symmetric solutions, negative saliency weights, and degenerate solutions). The use of O-INDSCAL (orthogonally constrained INDSCAL) was advocated in dealing with some of these issues. As demonstrated in Table 1, O-INDSCAL often provides solutions very similar to those obtained by original INDSCAL, yet completely avoids the degeneracy problem, and leads to algorithms with much faster convergence.

In this paper, an efficient algorithm has been proposed for O-INDSCAL by incorporating an acceleration technique called minimal polynomial extrapolation (MPE) into ten Berge et al.'s (1988) algorithm to accelerate the convergence. This algorithm is a first order algorithm (an algorithm that uses only the first order derivative information), and consequently, requires much less storage compared to such second order methods as the full- and quasi-Newton methods. It is also simple to implement, and easily extensible to other similar situations.

Simulation studies were conducted to demonstrate the efficiency of the proposed algorithm. In the simulation studies, three kinds of data sets were generated. One involved completely random data sets, and the other two involved structured data sets, one generated with indefinite weight matrices, and the other *nnd* weight matrices. In all cases, the MPE acceleration method showed a substantial reduction in computation time (roughly 50% to 70%) relative to ten Berge et al.'s original algorithm, and a more recently proposed algorithm by Trendafilov (2004). An optimal value of the degree of polynomials (j) in the MPE algorithm was found to be between 5 and 10. Within this interval, there was not very much difference in convergence speed. It should be noted, however, that no matter which value of j was chosen, the MPE algorithm reduced the mean CPU time significantly.

An efficient algorithm is essential in situations in which solutions have to be obtained repeatedly. As has been shown, INDSCAL is not free from the problem of suboptimal solutions. In some cases many solutions have to be obtained for the same data

set starting from many different initial values to ensure globally optimal solutions. Permutation tests for determining the number of significant dimensions (or components) also require repeated applications of the algorithm to data sets. These situations always call for more efficient algorithms.

The problem of convergence to suboptimal solutions was also investigated in the simulation studies. It turned out that the MPE method had little effect on reducing the chance of suboptimal solutions. The MPE algorithm produced suboptimal solutions as often as ten Berge et al's original algorithm. This is consistent with the observation made in similar contexts (Takane and Zhang, in press). Still, some interesting patterns of the difference between some algorithms emerged. The rate of suboptimal solutions was similar and rather high for all three algorithms when the model was "misspecified". The model was always misspecified for completely random data, and when the dimensionality was intentionally misspecified for structured data in some experiments. However, when the specified model is (nearly) correct, ten Berge et al's algorithm and the MPE algorithm encountered suboptimal solutions less than 2% of the time for both kinds of structured data, whereas Trendafilov's algorithm about 20% for structured data with indefinite weight matrices, which was significantly higher than the former. (Trendafilov's algorithm worked as well as the other algorithms for structured data with *nnd* weight matrices.) It was also shown that the number of runs from random initials needed to obtain a global minimum solution at least once could be estimated, based on the probability of a suboptimal solution in a single run.

It may be pointed out that the O-INDSCAL model subsumes an interesting special case. The CPC (Common Principal Component; Flury, 1988) model follows from the O-INDSCAL model when as many components (p) as there are variables (n) are extracted. Flury (1988) developed a maximum likelihood estimation procedure for fitting the CPC model, and Flury and Gautschi (1986; see also Clarkson, 1988) developed an algorithm for it, called the FG algorithm. De Leeuw and Pruzansky (1978) developed an algorithm called SUMSCAL (SUbjective Metric SCALing), which could be regarded as a least squares version of fitting the CPC model, although it was originally developed as a rational initialization procedure for the original INDSCAL model. SUMSCAL attempts to diagonalize several symmetric matrices simultaneously by repeated applications of simple plane rotations as in the Jacobi method for obtaining eigenvalues and vectors of a symmetric matrix. Carroll, De Soete, and Pruzansky (1989) compared several methods for initializing original INDSCAL. Ten Berge (1984) also developed an algorithm for simultaneous diagonalization of several symmetric matrices in view of the fact that certain orthogonal rotation problems in factor analysis could be reformulated in terms of simultaneous diagonalization problems. However, he found that his algorithm was inferior to SUMSCAL and Kaiser's (1958) original algorithm for Varimax. His algorithm may still be viable, however, with the incorporation of the MPE acceleration technique.

One important implication of the above algorithms is their potential use in deriving rational initial estimates for O-INDSCAL. Let $\mathbf{S}$ denote the mean of $\mathbf{S}_k$ across k. If the O-INDSCAL model is correct, $\mathbf{S}$ can be factored into:

$$\mathbf{S} = \mathbf{X}(\sum_{k=1}^{K} \mathbf{D}_k/K)\mathbf{X}' \tag{33}$$

by the spectral decomposition. Define

$$\mathbf{G}_k = \mathbf{X}'\mathbf{S}_k\mathbf{X} \tag{34}$$

for $k = 1, \cdots, K$. The matrices $\mathbf{G}_k$ may be simultaneously diagonalized by any of the algorithms mentioned in the previous paragraph to obtain an orthogonal matrix $\mathbf{T}$ (p by p) such that $\mathbf{XT}$ diagonalizes all the $\mathbf{G}_k$'s as much as possible. This rational initialization method may further speed up the convergence of the algorithms for O-INDSCAL.

The proposed MPE algorithm may readily be extended to other similar situations. Orthogonal IDIOSCAL (Individual DIfferences in Orientation SCALing; Carroll and Chang, 1970, 1972) is one such possibility. In the IDIOSCAL model, $\mathbf{D}_k$ in the INDSCAL model is replaced by $\mathbf{C}_k$, which is not necessarily diagonal, but only symmetric *nnd*. This allows idiosyncratic rotations of the matrix of common stimulus configuration by different individuals. Kiers (1989b) discusses how to constrain $\mathbf{C}_k$ to be *nnd* in the orthogonal IDIOSCAL algorithm he developed. With the help of this knowledge, it seems rather straightforward to extend the MPE algorithm to fit the orthogonal IDIOSCAL model.

The MPE algorithm may also be extended to the generalized GIPSCAL (Generalized Inner Product SCALing; Kiers and Takane, 1994) model. This model is a special case of DEDICOM (DEcomposition into DIrectional COMponents; Harshman, 1978), which is a model for the analysis of asymmetric square tables. Its three-way extension, 3-way GIPSCAL (Trendafilov, 2002), can be considered as a direct extension of INDSCAL/IDIOSCAL to asymmetric square tables. In 3-way GIPSCAL, the weight matrix (that corresponds to $\mathbf{D}_k$ in INDSCAL and $\mathbf{C}_k$ in IDIOSCAL) is an asymmetric square table that is assumed to be decomposed into two parts, an *nnd* diagonal matrix (like $\mathbf{D}_k$ in INDSCAL) plus a skew-symmetric matrix. When the skew-symmetric part is assumed null, this model reduces to the O-INDSCAL model. Again, it should not be difficult to extend the MPE algorithm to 3-way GIPSCAL. Trendafilov (2002) developed an algorithm for 3-way GIPSCAL based on the dynamical system approach. The comparison of the MPE algorithm with his algorithm in this context may be interesting.

6 Appendix

This appendix gives a detailed account of why the conditional estimate of one parameter set given the other can be treated as if it were constant when differentiating the conditional minimum function with respect to the second parameter set.

Let $\mathbf{A}$ and $\mathbf{B}$ be the two sets of parameters, and suppose we are minimizing $f(\mathbf{A}, \mathbf{B})$ with respect to these two sets of parameters. That is,

$$\min_{A,B} f(\mathbf{A}, \mathbf{B}). \tag{35}$$

We do this by

$$\min_{A,B} f(\mathbf{A}, \mathbf{B}) = \min_{A} \min_{B|A} f(\mathbf{A}, \mathbf{B}) = \min_{A} f(\mathbf{A}, \hat{\mathbf{B}}(\mathbf{A})), \tag{36}$$

where $f(\mathbf{A}, \hat{\mathbf{B}}(\mathbf{A})) = \min_{B|A} f(\mathbf{A}, \mathbf{B})$. For minimizing $f(\mathbf{A}, \hat{\mathbf{B}}(\mathbf{A}))$ we need its derivatives with respect to $\mathbf{A}$. However, this derivative can be obtained as if $\hat{\mathbf{B}}(\mathbf{A})$ were a constant, whereas in fact it is a function of $\mathbf{A}$. This is because $\hat{\mathbf{B}}(\mathbf{A})$ is obtained by minimizing $f(\mathbf{A}, \mathbf{B})$ with respect to $\mathbf{B}$ given $\mathbf{A}$. This will greatly simplify the derivation. A similar line of argument has been used for nonmetric multidimensional scaling

(Lingoes & Roskam, 1973), where stimulus coordinates and the best monotonic transformation of the proximity data that minimize a LS criterion have to be estimated simultaneously. To be more exact,

$$\begin{aligned}\frac{df(\mathbf{A},\hat{\mathbf{B}}(\mathbf{A}))}{d\mathbf{A}} &= \frac{\partial f(\mathbf{A},\hat{\mathbf{B}}(\mathbf{A}))}{\partial \mathbf{A}} + \frac{d\hat{\mathbf{B}}(\mathbf{A})}{d\mathbf{A}}\frac{\partial f(\mathbf{A},\hat{\mathbf{B}}(\mathbf{A}))}{\partial \hat{\mathbf{B}}(\mathbf{A})} \\ &= \frac{\partial f(\mathbf{A},\hat{\mathbf{B}}(\mathbf{A}))}{\partial \mathbf{A}}, \end{aligned} \tag{37}$$

since

$$\frac{\partial f(\mathbf{A},\hat{\mathbf{B}}(\mathbf{A}))}{\partial \hat{\mathbf{B}}(\mathbf{A})} = \mathbf{0}. \tag{38}$$

When the nonnegativity restrictions are imposed on $\mathbf{D}_k$,

$$\frac{\partial \phi_k(\mathbf{X},\mathbf{D}_k)}{\partial \mathbf{D}_k} = \mathbf{0} \tag{39}$$

may not hold at $\mathbf{D}_k = \hat{\mathbf{D}}_k$ for the original function $\phi_k(\mathbf{X},\mathbf{D}_k)$ to be minimized. In this case, the restrictions have to be explicitly incorporated into the function to be minimized. Let

$$\tilde{\phi}_k(\mathbf{X},\mathbf{D}_k) = \phi_k(\mathbf{X},\mathbf{D}_k) - 2\mathrm{tr}(\mathbf{M}_k\mathbf{D}_k), \tag{40}$$

where $\mathbf{M}_k$ is a diagonal matrix of Lagrangean multiplies such that

$$m_{ki} = 0, \quad \text{if} \quad d_{ki} > 0, \tag{41}$$

and

$$m_{ki} > 0, \quad \text{if} \quad d_{ki} = 0, \tag{42}$$

where m_{ki} and d_{ki} are the i^{th} diagonal elements of $\mathbf{M}_k$ and $\mathbf{D}_k$, respectively. The Karush-Kuhn-Tucker condition (e.g., Fletcher, 1981) states that

$$\frac{1}{2}\frac{\partial \tilde{\phi}_k(\mathbf{X},\mathbf{D}_k)}{\partial \mathbf{D}_k} = \mathrm{diag}(\mathbf{X}'\mathbf{S}_k\mathbf{X}) - \hat{\mathbf{D}}_k - \mathbf{M}_k = \mathbf{0} \tag{43}$$

is a necessary condition for a minimum, which leads to

$$\hat{\mathbf{D}}_k = \mathrm{diag}(\mathbf{X}'\mathbf{S}_k\mathbf{X}) - \mathbf{M}_k, \tag{44}$$

where

$$\mathbf{M}_k = \max(\mathbf{0}, -\mathrm{diag}(\mathbf{X}'\mathbf{S}_k\mathbf{X})). \tag{45}$$

Note that the term subtracted from $\phi_k(\mathbf{X},\mathbf{D}_k)$ in (40) has no effect in the estimation of $\mathbf{X}$, i.e., $\hat{\mathbf{D}}_k$ given in (44) can be considered constant for the purpose of updating $\mathbf{X}$.

Acknowledgements We would like to thank Ulf Böckenholt at McGill University for his insightful comments on an earlier draft of this paper, Sébastien Loisel at the University of Geneva for providing a five-line MATLAB routine for carrying out the minimal polynomial extrapolation (MPE), and Nicolay Trendafilov at Open University who kindly provided MATLAB codes for his algorithm.

References

Arabie P, Carroll JD, DeSarbo WS (1987) Three way scaling and clustering. Sage Publications, Newbury Park, CA

Carroll JD, Chang JJ (1970) Individual differences and multidimensional scaling via an N-way generalization of Eckart-Young decomposition. Psychometrika 35:282-319

Carroll JD, Chang JJ (1972) IDIOSCAL (Individual Differences In Orientation SCALing): A generalization of INDSCAL allowing IDIOsyncratic reference systems as well as an analytic approximation to INDSCAL. Paper presented at the Psychometric Society, Princeton, NJ

Carroll JD, De Soete G, Pruzansky S (1989) An evaluation of five algorithms for generating an initial configuration for SINDSCAL. J Classif 6:105-119

Clarkson DB (1988) A least squares version of algorithm AS 211: The F-G diagonalization algorithm. Appl Stat-J Roy St C 37:317-321

de Leeuw J, Pruzansky S. (1978) A new computational method to fit the weighted euclidean distance model. Psychometrika 43:479-490

Fletcher R (1981) Practical methods of optimization, Vol. 2, Constrained optimization. Wiley, Chichester

Flury B (1988) Common principal components and related multivariate models. Wiley, New York

Flury B, Gautschi W (1986) An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. SIAM J Sci Stat Comp 7:169-184.

Golub G, van Loan C (1996) Matrix computations, third edition. The Johns Hopkins University Press, London

Harshman RA (1978) Models for analysis of asymmetrical relationships among N objects or stimuli. Paper presented at the First Joint Meeting of the Psychometric Society and the Society of Mathematical Psychology, Hamilton, ON

Helm CE (1964) Multidimensional ratio scaling analysis of perceived color relations. J Opt Soc Am 54:256-262.

Jacobowitz D (1975) The acquisition of semantic structures. Unpublished doctoral dissertation, University of North Carolina.

Jennrich RI (2001) A simple general procedure for orthogonal rotation. Psychometrika 66:289-306

Kaiser HF (1958) The varimax criterion for analytic rotation in factor analysis. Psychometrika 23:187-200.

Kiers HAL (1989a) A computational short-cut for INDSCAL with orthonormality constraints on positive semi-definite matrices of low rank. Comput Stat Quart 2:119-135.

Kiers HAL (1989b) An alternating least squares algorithm for fitting the two- and three-way DEDICOM model and the IDIOSCAL model. Psychometrika 54:515-521

Kiers HAL (1991) Hierarchical relations among three-way methods. Psychometrika 56:449-470

Kiers HAL, Takane Y (1994) A generalization of GIPSCAL for the analysis of nonsymmetric data. J Classif 11:79-99

Krijnen WP, Dijkstra TK, Stegeman A (2008) On the non-existence of optimal solutions and the occurrence of "degeneracy" in the Candecomp/Parafac model. Psychometrika 73:431-439

Kroonenberg PM (1983) Three-mode principal component analysis. DSWO Press, Leiden.

Lingoes JC, Roskam EE (1973) A mathematical and empirical analysis of two multidimensional scaling algorithms. Psychometrika Monograph Supplement 19

Loisel S, Takane M (2009) Fast indirect robust generalized method of moments. Comput Stat and Data An 53:3571-3579

Nguyen TK, Oshima-Takane Y (2008) Do 2- to 3-year-old children use functional or shape cues to name objects? Poster presented at the International Society on Infant Studies, Vancouver, BC.

Rosenberg S, Kim MP (1975) The method of sorting as a data-gathering procedure in multivariate research. Multivar Behav Res 10:489-502

Schiffman SS, Reynolds ML, Young FW (1981) Introduction to Multidimensional Scaling. Academic Press, New York

Smith DA, Ford WF, Sidi A (1987) Extrapolation methods for vector sequences. SIAM Review 29:199-233

Stegeman A (2007) Degeneracy in Candecomp/Parafac explained for $p \times p \times 2$ arrays of rank $p + 1$ or higher. Psychometrika 71:483-501
Takane Y (2007) Applications of multidimensional scaling in psychometrics. In: Rao CR, Sinharay S (eds) Handbook of statistics (Vol. 26): Pyschometrics. Elsevier BV, Amsterdam, pp 359-400
Takane Y, Zhang Z (in press) Algorithms for DEDICOM: Acceleration, deceleration, or neither? J Chemometr
ten Berge JMF (1984) A joint treatment of varimax and the problem of diagonalizing symmetric matrices simultaneously in the least squares sense. Psychometrika 49:347-358
ten Berge JMF, Kiers HAL (1991) Some clarifiation of the CANDECOMP algorithm applied to INDSCAL. Psychometrika 56:317-326
ten Berge JMF, Kiers HAL, de Leeuw J (1988) Explicit candecomp/parafac solutions for a contrived $2 \times 2 \times 2$ array of rank three. Psychometrika 53:579-584
ten Berge JMF, Kiers HAL, Krijnen WP (1993) Computational solutions for the problem of negative saliences and nonsymmetry in INDSCAL. J Classif 10:115-124
ten Berge JMF, Knol DL, Kiers HAL (1988) A treatment of the orthomax rotation family in terms of diagonalization, and a re-examination of a singular value approach to varimax rotation. Comput Stat Quart 3:207-217
Trendafilov N (2002) GIPSCAL revisited: A projected gradient approach. Stat Comput 12:135-145
Trendafilov N (2004) Orthonormality-constrained INDSCAL with nonnegative salience. In: Legan A et al. (eds) Computational science and its applications. Lecture Notes in Computer Science 3044, Part II. Springer, Berlin, pp 952-960
Zhang Z, Takane Y (in press) Statistics: Multidimensional scaling. In: Baker E, McGaw B, Peterson PP (eds) International encyclopedia of education, 3rd edn. Elsevier, Oxford