Algorithms for DEDICOM: Acceleration, Deceleration, or Neither?

Yoshio Takane, McGill University

Zhidong Zhang, McGill University

We would like to thank Sébastien Loisel for providing a five-line MATLAB routine for carrying out the minimal polynomial extrapolation (MPE).

Abstract

Takane's original algorithm for DEDICOM (DEcomposition into DIrectional COMponents) was proposed more than two decades ago. There have been a couple of significant developments since then: Kiers et al.'s modification to ensure monotonic convergence of the algorithm, and Jennrich's recommendation to use the modified algorithm only when Takane's original algorithm violates the monotonicity. In this paper, we argue that neither of these modifications is essential, drawing a close relationship between Takane's algorithm and the simultaneous power method for obtaining dominant eigenvalues and vectors of a symmetric matrix. By ignoring monotonicity, we can develop a much more efficient algorithm by simple modifications of Takane's original algorithm, as demonstrated in this paper. More specifically, we incorporate the minimum polynomial extrapolation (MPE) method to accelerate the convergence of Takane's algorithm, and show that it significantly cuts down the computation time.

1 Takane's personal note

Richard Harshman left us two important legacies. This paper is closely related to one of them. At the 1978 meeting of the Psychometric Society in Hamilton, Ontario, Richard presented a seminal paper on DEDICOM, a model for the analysis of asymmetric square tables (Harshman, 1978). His paper opened up a new horizon for asymmetric data analysis, which until then drew little attention from statisticians. People were, however, quick to recognize the importance of the topic in statistics. Only two years later, a one-day workshop on asymmetric data analysis was organized through the initiative of Joe Kruskal and Richard himself. This satellite meeting to the annual meeting of the Psychometric Society in Iowa attracted over 30 people from around the world. One of the authors of the present paper (Takane) was invited to give a talk at the workshop, and he initially accepted the invitation in the hope that he would be able to come up with something worth presenting. However, despite a great deal of effort, even one week before the workshop he was still unable to think of anything worthwhile, and decided to withdraw his talk. This turned out to be a nightmare for him in dual sense. First of all, it was disgraceful to withdraw a paper shortly before the meeting, and secondly, just a few days after he withdrew his talk, he suddenly came up with an algorithm, which is now known as Takane's algorithm for DEDICOM. He missed the chance of presenting his algorithm at the workshop, although a short paper (Takane, 1985) describing the algorithm came out several years after the workshop. The present paper describes some recent developments in this algorithm.

2 Introduction

Numerous situations give rise to square asymmetric tables. Mobility tables, journal citation data, brand loyalty data, and stimulus identification data are but a few examples of such tables. Two-way single-domain DEDICOM (DEcomposing DIrectional COMponents; Harshman, 1978; Harshman, Green, Wind, and Lundy, 1981) is a model for the analysis of such asymmetric square tables. Let $\tilde{\mathbf{A}}$ represent a square asymmetric table of size n. The infallible DEDICOM model of dimensionality r is written as

$$\tilde{\mathbf{A}} = \mathbf{X}\mathbf{B}\mathbf{X}', \tag{1}$$

where $\mathbf{X}$ is an n by r columnwise nonsingular matrix such that $\text{Sp}(\mathbf{X}) = \text{Sp}(\tilde{\mathbf{A}}) = \text{Sp}(\tilde{\mathbf{A}}')$, and $\mathbf{B}$ a nonsingular matrix of order r. The DEDICOM model attempts to explain asymmetric relationships among n objects by asymmetric relationships among a smaller number (r) of latent "objects".

Takane (1985) developed a least squares (LS) method for fitting the DEDICOM model to fallible data, which, although iterative, turned out to be quite efficient in many practical situations. However, his algorithm is not always monotonically convergent in the sense that it consistently minimizes the value of the LS criterion. A modified algorithm has been proposed (Kiers, ten Berge, Takane, and de Leeuw, 1990) that ensures monotonic convergence. However, this modification tends to slow the speed of convergence of the algorithm. Consequently, Jennrich (2001) proposed to use the modified algorithm only when the monotonicity is actually violated. Following his strategy, the algorithm maintains monotonicity in convergence, and a substantial improvement has been gained in convergence speed. In this paper, we argue that the monotonicity in convergence is not a crucial property of an algorithm, drawing a close relationship between Takane's algorithm for DEDICOM and the simultaneous power method for obtaining dominant eigenvalues and vectors of a symmetric matrix. The latter, like the former, has no monotonic convergence property unless the matrix is *nnd* (nonnegative definite). We show that Takane's algorithm can be further improved in convergence speed by ignoring monotonicity, and incorporating some acceleration techniques developed in the context of simultaneous power iterations.

3 The DEDICOM model

In this section, we discuss the DEDICOM model in some detail. We first take up the error-free case, and then the error-contaminated case. In the former, we focus on algebraic properties of the model, and in the latter, estimating model parameters.

3.1 Infallible case

A square matrix $\tilde{\mathbf{A}}$ of order n and of rank r is called an EP$_r$ matrix if and only if

$$\mathrm{Sp}(\tilde{\mathbf{A}}) = \mathrm{Sp}(\tilde{\mathbf{A}}'), \tag{2}$$

where $\mathrm{Sp}(\tilde{\mathbf{A}})$ indicates the range (column) space of $\tilde{\mathbf{A}}$. The name EP$_r$ was first introduced by Schwerdtfeger (1950), who defined it as a set of square matrices $\tilde{\mathbf{A}}$ of rank r such that $\tilde{\mathbf{A}}\mathbf{x} = \mathbf{0}$ if and only if $\tilde{\mathbf{A}}'\mathbf{x} = \mathbf{0}$. That is, $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{A}}'$ have the same null space, or equivalently the same range space. Condition (2) is sometimes called range symmetry, and generalizes the usual notion of symmetry in which $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}'$. Special cases of EP matrices include symmetric matrices, skew-symmetric matrices, normal matrices ($\tilde{\mathbf{A}}'\tilde{\mathbf{A}} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}'$), and so on.

Note that square nonsingular matrices of order n are always EP$_n$, and that EP$_1$ matrices are always symmetric.

There are a number of interesting properties of EP matrices (e.g., Tian, 2008), of which the one most relevant to the present paper is noted here. EP$_r$ matrices admit a representation by the DEDICOM model given in (1). In fact, there is one-to-one correspondence between an EP$_r$ matrix of order n and the DEDICOM representation. Given the representation (1), it is obvious that $\text{Sp}(\tilde{\mathbf{A}}) = \text{Sp}(\tilde{\mathbf{A}}') \supset \text{Sp}(\mathbf{X})$. By postmultiplying both sides of (1) by $\mathbf{XB}'(\mathbf{BX}'\mathbf{XB}')^{-1}$, we obtain $\tilde{\mathbf{A}}\mathbf{XB}'(\mathbf{BX}'\mathbf{XB}')^{-1} = \mathbf{X}$, which indicates $\text{Sp}(\mathbf{X}) \supset \text{Sp}(\tilde{\mathbf{A}})$. That $\text{Sp}(\mathbf{X}) \supset \text{Sp}(\tilde{\mathbf{A}}')$ can be similarly shown. Conversely, let $\mathbf{X}$ be an n by r columnwise nonsingular matrix consisting of basis vectors spanning $\text{Sp}(\tilde{\mathbf{A}})$. Then, $\tilde{\mathbf{A}} = \mathbf{XC}$ for some rowwise nonsingular matrix $\mathbf{C}$, but $\text{Sp}(\tilde{\mathbf{A}}) = \text{Sp}(\tilde{\mathbf{A}}')$ implies $\mathbf{C}' = \mathbf{XB}'$ or $\mathbf{C} = \mathbf{BX}'$ for some nonsingular matrix $\mathbf{B}$ of order r. Combining the two, we obtain $\tilde{\mathbf{A}} = \mathbf{XBX}'$.

The above result indicates that the infallible DEDICOM model requires the EP$_r$-ness of matrix $\tilde{\mathbf{A}}$ of order n. The representation (1) is not unique, however, because the set of basis vectors spanning $\text{Sp}(\tilde{\mathbf{A}}) = \text{Sp}(\tilde{\mathbf{A}}')$ is not unique. For convenience, we require that $\mathbf{X}$ is columnwise orthonormal. If $\mathbf{X}$ is not orthonormal initially, it can always be made so. Let $\mathbf{X} = \mathbf{PDQ}'$ denote the singular value decomposition (SVD) of $\mathbf{X}$. Then, $\tilde{\mathbf{A}} = \mathbf{PDQ}'\mathbf{BQDP}'$. We may then set $\mathbf{X}_{new} = \mathbf{P}$, and $\mathbf{B}_{new} = \mathbf{DQ}'\mathbf{BQD}$. The columnwise orthonormality of $\mathbf{X}$ is still not sufficient to make the representation (1) unique because there are infinitely many ways to define orthonormal bases for $\text{Sp}(\tilde{\mathbf{A}})$. To make the DEDICOM representation completely unique (except for reflections and permutations of the columns of $\mathbf{X}$), we further require

$$\mathbf{BB}' + \mathbf{B}'\mathbf{B} = \boldsymbol{\Delta} \quad \text{(diagonal and positive definite } (pd)\text{)}. \tag{3}$$

If the original $\mathbf{B}$ fails to satisfy this condition, it can always be transformed into one that satisfies the condition. Let $\mathbf{BB}' + \mathbf{B}'\mathbf{B} = \mathbf{T}\tilde{\boldsymbol{\Delta}}\mathbf{T}'$ denote the spectral decomposition of $\mathbf{BB}' + \mathbf{B}'\mathbf{B}$. Then, the new $\mathbf{X}$ and $\mathbf{B}$ can be obtained by $\mathbf{X}_{new} = \mathbf{XT}$, and $\mathbf{B}_{new} = \mathbf{T}'\mathbf{BT}$, respectively. It is obvious that $\mathbf{X}_{new}$ remains columnwise orthogonal, and that

$\mathbf{B}_{new}\mathbf{B}'_{new} + \mathbf{B}'_{new}\mathbf{B}_{new} = \tilde{\mathbf{\Delta}} \equiv \mathbf{\Delta}$ is satisfied.

There is a closed form solution to the infallible DEDICOM model (i.e., an exact way to obtain $\mathbf{X}$ and $\mathbf{B}$ from $\tilde{\mathbf{A}}$). Let

$$\tilde{\mathbf{A}}\tilde{\mathbf{A}}' + \tilde{\mathbf{A}}'\tilde{\mathbf{A}} = \tilde{\mathbf{P}}\tilde{\mathbf{D}}\tilde{\mathbf{P}}' \tag{4}$$

be the spectral decomposition of $\tilde{\mathbf{A}}\tilde{\mathbf{A}}' + \tilde{\mathbf{A}}'\tilde{\mathbf{A}}$, where $\tilde{\mathbf{P}}$ is the n by r matrix of eigenvectors, and $\tilde{\mathbf{D}}$ the r by r positive diagonal matrix of eigenvalues. Then, $\mathbf{X} = \tilde{\mathbf{P}}$ and $\mathbf{B} = \mathbf{X}'\tilde{\mathbf{A}}\mathbf{X}$ solve (1) exactly. It can be easily seen that

$$\mathbf{B}\mathbf{B}' + \mathbf{B}'\mathbf{B} = \tilde{\mathbf{P}}'(\tilde{\mathbf{A}}\tilde{\mathbf{A}}' + \tilde{\mathbf{A}}'\tilde{\mathbf{A}})\tilde{\mathbf{P}} = \tilde{\mathbf{D}}(\equiv \mathbf{\Delta}) \quad \text{(diagonal).} \tag{5}$$

(Note that $\tilde{\mathbf{A}}\mathbf{X}\mathbf{X}'\tilde{\mathbf{A}}' = \tilde{\mathbf{A}}\tilde{\mathbf{A}}'$, and $\tilde{\mathbf{A}}'\mathbf{X}\mathbf{X}'\tilde{\mathbf{A}} = \tilde{\mathbf{A}}'\tilde{\mathbf{A}}$.)

There are two important special cases that deserve particular attention: (1) $\tilde{\mathbf{A}}$ is symmetric, and (2) $\tilde{\mathbf{A}}$ is skew-symmetric ($\tilde{\mathbf{A}}' = -\tilde{\mathbf{A}}$). When $\tilde{\mathbf{A}}$ is symmetric, $\tilde{\mathbf{A}}\tilde{\mathbf{A}}' + \tilde{\mathbf{A}}'\tilde{\mathbf{A}}$ reduces to $2(\tilde{\mathbf{A}})^2$. The DEDICOM model in this case simply reduces to the spectral decomposition of matrix $\tilde{\mathbf{A}}$. When $\tilde{\mathbf{A}}$ is skew-symmetric, let $\tilde{\mathbf{A}} = \tilde{\mathbf{P}}(\tilde{\mathbf{D}}^{1/2}/\sqrt{2})\tilde{\mathbf{Q}}'$ be the SVD of $\tilde{\mathbf{A}}$. Nonzero singular values of a skew-symmetric matrix come in pairs, and $\tilde{\mathbf{P}}(\tilde{\mathbf{D}}^{1/2}/\sqrt{2})\tilde{\mathbf{Q}}'$ can be further rewritten as $\tilde{\mathbf{P}}(\tilde{\mathbf{D}}^{1/2}/\sqrt{2})\tilde{\mathbf{Q}}' = \tilde{\mathbf{P}}(\tilde{\mathbf{D}}^{1/2}/\sqrt{2})\mathbf{L}\tilde{\mathbf{P}}' = \mathbf{X}\mathbf{B}\mathbf{X}'$ by setting $\mathbf{X} = \tilde{\mathbf{P}}$, and $\mathbf{B} = (\tilde{\mathbf{D}}^{1/2}/\sqrt{2})\mathbf{L}$, where $\mathbf{L}$ is a block diagonal matrix consisting of 2 by 2 blocks of the form $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. It can be easily verified that $\mathbf{B}\mathbf{B}' + \mathbf{B}'\mathbf{B} = \tilde{\mathbf{D}}$ because $\mathbf{L}'\mathbf{L} = \mathbf{L}\mathbf{L}' = \mathbf{I}$. This case is equivalent to Canonical Analysis of a SKew-symmetric (CASK) matrix by Gower (1977; see also Takane, 2004). Note that r is always even for a skew-symmetric matrix.

3.2 Fallible case: Takane's algorithm for DEDICOM and its modifications

In most practical data analysis situations, the data matrix $\mathbf{A}$ is not likely to be exactly EP$_r$, and so we approximate $\mathbf{A}$ by an EP$_r$ matrix. That is,

$$\mathbf{A} = \tilde{\mathbf{A}} + \mathbf{E}, \tag{6}$$

where $\mathbf{E}$ is the matrix of disturbance terms, and we attempt to find $\mathbf{X}$ and $\mathbf{B}$ that minimize the least squares (LS) criterion,

$$f(\mathbf{X}, \mathbf{B}) = \mathrm{tr}(\mathbf{E}'\mathbf{E}) \tag{7}$$

subject to the identification restrictions

$$\mathbf{X}'\mathbf{X} = \mathbf{I}_r, \tag{8}$$

and

$$\mathbf{B}\mathbf{B}' + \mathbf{B}'\mathbf{B} = \boldsymbol{\Delta}. \tag{9}$$

The minimum of f with respect to $\mathbf{B}$ is obtained by

$$\mathbf{B} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{A}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \mathbf{X}'\mathbf{A}\mathbf{X}. \tag{10}$$

(The second equality holds only when $\mathbf{X}$ is columnwise orthonormal.) If we put this estimate of $\mathbf{B}$ in (7), we obtain

$$f^*(\mathbf{X}) = \min_B f(\mathbf{X}, \mathbf{B}) = \mathrm{tr}(\mathbf{A}'\mathbf{A}) - \mathrm{tr}(\mathbf{B}'\mathbf{B}). \tag{11}$$

Since $\mathrm{tr}(\mathbf{A}'\mathbf{A})$ is constant once the data are given, minimizing $f^*(\mathbf{X})$ with respect to $\mathbf{X}$ subject to (8) is equivalent to maximizing

$$g(\mathbf{X}) = \frac{1}{2}\mathrm{tr}(\mathbf{B}'\mathbf{B}) \tag{12}$$

under the same restriction. (The factor of 1/2 preceding $\mathrm{tr}(\mathbf{B}'\mathbf{B})$ is merely for convenience.)

This constrained maximization problem can be turned into an unconstrained maximization problem by introducing Lagrange multipliers and defining a new function. Let $\mathbf{S}$ denote a symmetric matrix of Lagrange multipliers, and define

$$g^*(\mathbf{X}, \mathbf{S}) = g(\mathbf{X}) - \frac{1}{2}\mathrm{tr}(\mathbf{S}(\mathbf{X}'\mathbf{X} - \mathbf{I})). \tag{13}$$

We maximize this function with respect to $\mathbf{X}$. Differentiating $g^*(\mathbf{X}, \mathbf{S})$ with respect to $\mathbf{X}$ and $\mathbf{S}$, and setting the results equal to zero, we obtain

$$\mathbf{A}\mathbf{X}\mathbf{B}' + \mathbf{A}'\mathbf{X}\mathbf{B} - \mathbf{X}\mathbf{S} = \mathbf{0}, \tag{14}$$

and

$$\mathbf{X}'\mathbf{X} - \mathbf{I} = \mathbf{0}. \tag{15}$$

Let

$$\frac{\partial g(\mathbf{X})}{\partial \mathbf{X}} \equiv \mathbf{G} = \mathbf{AXB}' + \mathbf{A}'\mathbf{XB}. \tag{16}$$

Then, premultiplying (14) by $\mathbf{X}'$, we obtain $\mathbf{X}'\mathbf{G} = \mathbf{BB}' + \mathbf{B}'\mathbf{B} = \mathbf{S}$ under (8). (This confirms that $\mathbf{S}$ is indeed symmetric.) Constraint (9) is not yet explicitly imposed. To do so, let $\mathbf{S} = \mathbf{T\Delta T}'$ denote the spectral decomposition of $\mathbf{S}$. Then, postmultiplying (14) by $\mathbf{T}$, we obtain $\mathbf{AXB}'\mathbf{T} + \mathbf{A}'\mathbf{XBT} = \mathbf{AXTT}'\mathbf{X}'\mathbf{A}'\mathbf{XT} + \mathbf{A}'\mathbf{XTT}'\mathbf{X}'\mathbf{AXT} = \mathbf{XT\Delta}$. If we take $\mathbf{XT}$ as new $\mathbf{X}$ in this equation, we obtain

$$\mathbf{AXB}' + \mathbf{A}'\mathbf{XB} = \mathbf{X\Delta} \tag{17}$$

to be solved. This looks like an eigen-equation, but unfortunately it is not, because $\mathbf{B} = \mathbf{X}'\mathbf{AX}$ is also a function of $\mathbf{X}$. It is in fact a cubic equation in $\mathbf{X}$, and in general it is difficult to solve in closed form.

Takane (1985) proposed to solve equation (17) by applying the following two steps iteratively:

Step I: Calculate $\mathbf{X}^* = \mathbf{G} = \mathbf{AXB}' + \mathbf{A}'\mathbf{XB}$.

Step II: Apply the Gram-Schmidt orthonormalization to $\mathbf{X}^*$ to obtain a new $\mathbf{X}$, and go back to Step I.

This algorithm is called Takane's algorithm for DEDICOM. (Takane's algorithm can be interpreted as regarding $\mathbf{B}$ as constant temporarily, and solving the resultant linear equations. However, because $\mathbf{B}$ is in fact not constant, iterative adjustments to $\mathbf{X}$ are needed.)

The second step decomposes $\mathbf{X}^*$ into $\mathbf{XR}$, and defines $\mathbf{X} = \mathbf{X}^*\mathbf{R}^{-1}$, where $\mathbf{R}$ is an upper triangular matrix. At a convergence point, the equation

$$\mathbf{AXB}' + \mathbf{A}'\mathbf{XB} = \mathbf{XR} \tag{18}$$

must hold exactly. By premultiplying (18) by $\mathbf{X}'$, we obtain $\mathbf{B}\mathbf{B}' + \mathbf{B}'\mathbf{B} = \mathbf{R}$. Since the left hand side of this equation is symmetric, and $\mathbf{R}$ is upper triangular, $\mathbf{R}$ must be diagonal. (The only symmetric matrix which is also upper triangular is diagonal.) This shows that both (8) and (9) are satisfied at a convergence point. Step II above may be replaced by:

Step II': Obtain the SVD of $\mathbf{X}^*$, i.e., $\mathbf{X}^* = \mathbf{P}\mathbf{D}\mathbf{Q}'$. Then set $\mathbf{X} = \mathbf{P}$ and go back to Step I.

Step II' makes the algorithm converge slightly faster.

Let $\mathbf{P}_r\mathbf{D}_r\mathbf{P}_r'$ denote the best rank r approximation to $\mathbf{A}\mathbf{A}' + \mathbf{A}'\mathbf{A}$. Then $\mathbf{X} = \mathbf{P}_r$ gives a good initial estimate of $\mathbf{X}$ to start Takane's algorithm described above. It considerably reduces the number of iterations to convergence. It also helps avoid convergence to nonglobal optima, although the problem of nonglobal optima is not completely resolved by this convention. It is still recommended to obtain multiple solutions starting from multiple initial estimates to ensure a globally optimal solution in each case.

Takane's algorithm can be viewed as a special case of the gradient projection method (Jennrich, 2001; Edelman, Arias, and Smith, 1998). The gradient of $g(\mathbf{X})$ with respect to $\mathbf{X}$ is given by $\mathbf{G}$ as defined in (16), which is projected onto the tangent space of the Stiefel manifold (the set of all columnwise orthonormal matrices) at $\mathbf{X}$. This projection is given by

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

We attempt to make this projected gradient as close to zero as possible. (Note that the first term on the left hand side of the above formula is identically equal to $\mathbf{0}$ because $\mathbf{X}'\mathbf{G}$ is always symmetric in the present case.) Jennrich (2001) suggested that the Frobenius norm of the projected gradient be used as the convergence criterion. This criterion is attractive because it does not depend on the stepsize often associated with the updating equations for parameter estimates. We use

$$||(\mathbf{I} - \mathbf{X}\mathbf{X}')\mathbf{G}|| \leq 10^{-6} \tag{20}$$

in all numerical results to be reported subsequently, where $||\mathbf{F}|| = (\text{tr}(\mathbf{F}'\mathbf{F}))^{1/2}$ indicates the Frobenius norm of matrix $\mathbf{F}$. Note, however, that the norm on the left hand side of (20) is not independent of the scale unit used to collect the data matrix $\mathbf{A}$, and the right hand side may have to be adjusted accordingly.

Takane's algorithm turned out to be not monotonically convergent in the sense that it consistently maximizes (12) (or equivalently, minimizes (7)), except when $\mathbf{A} + \mathbf{A}'$ is *nnd* and of at least rank r. Kiers et al. (1990), however, showed, based on the majorization principle for finding a maximum of a matrix trace function (Kiers, 1990), that modifying Step I of Takane's algorithm by

$$\mathbf{X}^* = \mathbf{G} + \alpha\mathbf{X}, \tag{21}$$

where α is a prescribed constant, will guarantee monotonic convergence. The constant α should be at least as large as the largest eigenvalue of $-[(\mathbf{A} \otimes \mathbf{B}) + (\mathbf{A} \otimes \mathbf{B})']$ (where $\otimes$ indicates a Kronecker product). This α depends on $\mathbf{X}$ through $\mathbf{B}$, and consequently has to be updated every time $\mathbf{X}$ is updated. A sufficiently large value of α that does not depend on $\mathbf{X}$ is given by the largest eigenvalue of $-[(\mathbf{A} \otimes \mathbf{A}) + (\mathbf{A} \otimes \mathbf{A})']$, or twice the square of the largest singular value of $\mathbf{A}$. We call the modified algorithm by Kiers et al. (1990) the KBTL algorithm.

When $\mathbf{A}$ is *pd*, no modification to Takane's algorithm is necessary to ensure monotonic convergence ($\alpha = 0$ is sufficient for monotonic convergence in this case), but a negative α may still accelerate the convergence (without hampering the monotonicity). When the value of α is positive, the KBTL algorithm can considerably slow the convergence speed. Jennrich (2001) consequently recommends the use of the KBTL modification only when Takane's original algorithm exhibits a violation of monotonic convergence. The KBTL procedure only gives a sufficient condition for monotonic convergence, and Takane's algorithm may not violate monotonicity of convergence, even if the KBTL procedure points to a positive value of α. Jennrich's procedure is monotonic because it switches to the KBTL procedure as soon as Takane's algorithm violates the monotonicity. He shows a substantial improvement in the speed of convergence using his procedure. In his demonstration, many 10 by 10 matrices were generated with each of the elements following an independent standard normal distribution, and 5-component DEDICOM solutions were obtained. Jennrich (2001) found that his recommended procedure was about four times faster than the KBTL algorithm with the third choice of α described above (i.e., twice the square of the largest singular value of $\mathbf{A}$). Takane's algorithm, ignoring the monotonicity in convergence, is presumed to be at least as fast as Jennrich's procedure.

4 Similarity of Takane's algorithm to the simultaneous power iterations

For the special case of a symmetric matrix $\mathbf{A}$, Takane's algorithm for DEDICOM is similar to the simultaneous power method for obtaining r dominant eigenvalues and vectors of $\mathbf{A}$. In the simplest version of the simultaneous power method (Bauer, 1957), $\mathbf{X}$ is updated by first calculating (1) $\mathbf{X}^* = \mathbf{A}\mathbf{X}^{(q)}$, which is then (2) orthonormalized by $\mathbf{X}^{(q+1)} = \mathbf{X}^*\mathbf{R}^{-1}$, where $\mathbf{R}$ is the upper triangular matrix obtained by the Cholesky factorization of $\mathbf{X}^{*\prime}\mathbf{X}^*$, which is equivalent to that obtained by the Gram-Schmidt orthonormalization of $\mathbf{X}^*$. These two steps are analogous to Steps I and II in Takane's algorithm for DEDICOM.

Rutishauser (1969) proposed a slightly different procedure which uses an orthonormalization step analogous to Step II' in Takane's algorithm. In this procedure, the second step above is replaced by the spectral decomposition of $\mathbf{X}^{*\prime}\mathbf{X}^* = \mathbf{X}'\mathbf{A}^2\mathbf{X}$. Let $\mathbf{Q}\mathbf{D}^2\mathbf{Q}'$ denote the spectral decomposition of $\mathbf{X}^{*\prime}\mathbf{X}^*$. Then, $\mathbf{X}^{(q+1)}$ is obtained by $\mathbf{X}^*\mathbf{Q}\mathbf{D}^{-1}$, which is equivalent to setting $\mathbf{X}^{(q+1)} = \mathbf{P}$, where $\mathbf{P}$ is obtained by the SVD of $\mathbf{X}^* = \mathbf{P}\mathbf{D}\mathbf{Q}'$. Under some mild conditions ($|\lambda_r| > |\lambda_{r+1}|$, where λ_r is the r^{th} most dominant eigenvalue of $\mathbf{A}$ in the absolute value, and none of the column vectors in the initial $\mathbf{X}$ are orthogonal to the eigenvectors to be obtained), both Bauer's and Rutishauser's algorithms have been shown to converge to a set of eigenvectors corresponding to the r dominant eigenvalues of $\mathbf{A}$.

Takane's algorithm for DEDICOM, on the other hand, obtains the Gram-Schmidt orthonormalization of $2\mathbf{AX}(\mathbf{X}'\mathbf{AX})$ when $\mathbf{A}$ is symmetric. Since the postmultiplication of $\mathbf{AX}$ by $\mathbf{X}'\mathbf{AX}$ does not change the column space of $\mathbf{AX}$, Takane's algorithm converges to the same solution as both versions of the simultaneous power method described above, although it is slightly less efficient due to the additional and unnecessary computations involved for a symmetric matrix $\mathbf{A}$.

A number of acceleration techniques have been suggested for the simultaneous power method, and the similarity of the two algorithms suggests that some of them may work in the context of Takane's algorithm for DEDICOM. One recommended procedure in the simultaneous power method is to apply the orthonormalization step as few times as possible during the course of iterations. That is, instead of applying Step II of Takane's algorithm in every iteration, it is done, for example, every 5 or 10 iterations. By avoiding the additional computations involved in the orthonormalization step, this strategy is expected to save some computation time, although in DEDICOM, $\mathbf{X}^{*\prime}\mathbf{X}^*$ has to be inverted in order to calculate $\mathbf{B}$ (because $\mathbf{X} = \mathbf{X}^*$ is not orthonormal in the iterations in which the orthonormalization phase is skipped), which might take almost as much time as calculating $\mathbf{R}^{-1}$ in orthonormalization. Somewhat unexpectedly, however, this strategy turned out to work very well in Takane's algorithm according to the numerical experiment carried out in a similar set-up to Jennrich's (2001) experiment.

In this study, three sets of one hundred data matrices of varying sizes ($n = 10, 50, 100$) were randomly generated according to an independent standard normal distribution for each entry. DEDICOM solutions were obtained for each data matrix with selected numbers of components (for $n = 10$, the number of components (r) was taken to be either 3 or 5, and for $n = 50$ and $n = 100$, r was taken to be either 5 or 10) using the two algorithms: Takane's original algorithm and one that implements the non-orthonormalization strategy in several consecutive iterations. In the latter, the orthonormalization was carried out every 5 iterations or 10 iterations. In all cases, uniform random numbers initiated the iterative process. Table 1 shows the mean cpu time required for convergence over the 100 replicated data matrices. As can be seen, the cpu time is reduced by roughly 10% to 40% by not orthonormalizing $\mathbf{X}$ in every iteration. This is a substantial gain in efficiency over Takane's original algorithm, which is reputed to be quite efficient already. The number of consecutive iterations in which no orthonormalizations are applied does not seem to have substantial effects on the cpu time beyond 5. (The case of orthonormalization in every 15 iterations was also tried, but not reported in Table 1 because the cpu time was so similar to that for the 10 iterations.)

***** Insert Table 1 about here. *****

One important remark to be made is that in both Takane's algorithm and its modification above, the monotonicity principle is ignored. This is partly justified by the fact that neither Bauer's algorithm nor Rutishauser's algorithm possesses the monotonicity property with respect to $\frac{1}{2}\mathrm{tr}((\mathbf{X}'\mathbf{A}\mathbf{X})^2)$ that they are presumed to maximize.

5 Some acceleration/deceleration techniques by stepsize adjustments

Ramsay (1975) proposed an acceleration/deceleration technique applicable to a wide class of iterative parameter estimation methods. In his method, $\mathbf{X}$ is updated by

$$\mathbf{X}^* = (1-\beta)\mathbf{X}^{(q+1)} + \beta\mathbf{X}^{(q)}, \tag{22}$$

where q indicates the iteration number, and β is the parameter that regulates the convergence speed. In Takane's and the KBTL algorithms, $\mathbf{X}^{(q+1)}$ (the $(q+1)^{th}$ update of $\mathbf{X}$ using an unaccelerated method) is taken to be $\mathbf{G}$ defined in (16). When $\beta < 0$, an acceleration occurs (with $\mathbf{X}^*$ being further away from the previous $\mathbf{X}^{(q)}$), while a deceleration occurs when $1 > \beta > 0$ (with $\mathbf{X}^*$ being more heavily influenced by the previous estimate $\mathbf{X}^{(q)}$ than its new update $\mathbf{X}^{(q+1)}$). An acceleration is needed when the unaccelerated update ($\beta = 0$) approaches the convergence point too slowly, while a deceleration is needed when the standard update approaches the convergence point too "quickly", and toggles back and forth around the target point. An optimal value of β is determined by observing three consecutive updates of $\mathbf{X}$ according to

$$\beta' = 1 - (1-\beta)\frac{||\mathbf{U}^{(q)}||}{||\mathbf{V}^{(q)}||}, \tag{23}$$

where β' is the new stepsize, β the old stepsize, $\mathbf{U}^{(q)} = \mathbf{X}^{(q+1)} - \mathbf{X}^{(q)}$ (the first order difference), and $\mathbf{V}^{(q)} = \mathbf{U}^{(q+1)} - \mathbf{U}^{(q)} = \mathbf{X}^{(q+2)} - 2\mathbf{X}^{(q+1)} + \mathbf{X}^{(q)}$ (the second order difference). The value of β is initially taken to be zero.

Both Takane's and the KBTL algorithms are special cases of (22). To establish the exact relationships, we modify (22) slightly:

$$\mathbf{X}^* = \gamma(1-\beta)\mathbf{G} + \gamma\beta\mathbf{X}^{(q)}, \tag{24}$$

where an additional parameter $\gamma > 0$ has been introduced. This parameter has no effect on the final estimate of $\mathbf{X}$ because $\mathbf{X}^*$ is orthonormalized in Step II anyway which nullifies its effect. Takane's algorithm ($\mathbf{X}^* = \mathbf{G} = \mathbf{G} + (0)\mathbf{X}^{(q)}$), follows from (24) by setting $\gamma = 1$, and $\beta = 0$. Similarly, the KBTL algorithm ($\mathbf{X}^* = \mathbf{G} + \alpha\mathbf{X}^{(q)} = (1)\mathbf{G} + \alpha\mathbf{X}^{(q)}$)) follows from (24) by setting $\gamma\beta = \alpha$, and $\gamma(1-\beta) = 1$, that is, $\gamma = 1 + \alpha$, and $\beta = \alpha/(1+\alpha)$.

It is interesting to note that the steepest ascent algorithm $\mathbf{X}^* = \delta\mathbf{G} + \mathbf{X}^{(q)}$, which is a general iterative algorithm that could have been used in DEDICOM, can also be considered as a special case of (24). This can be seen by setting $\gamma\beta = 1$, and $\gamma(1-\beta) = \delta$ in (24), that is, $\gamma = 1 + \delta$, and $\beta = 1/(1+\delta)$.

These relations prompt us to apply Ramsay's acceleration/deceleration technique to DEDICOM with "optimal" updates of stepsize parameters. In this study, the same data sets that were used to obtain Table 1 were reanalyzed by Ramsay's method. The results were rather mixed, however. It sometimes reduced the cpu time for convergence significantly, but it also sometimes increased the cpu time significantly. Overall, there was no significant change in the cpu time. We conjecture that this rather mixed performance of Ramsay's acceleration procedure is due to the fact that it looks at only three consecutive iterations.

The MPE method to be discussed in the next section (and which turns out to be quite successful) looks at 10 to 15 successive iterations for best performance.

6 The minimal polynomial extrapolation (MPE) method

In this section, we explore a different type of acceleration technique, based on solving polynomial equations. For notational convenience, we use the vectorized form of parameter matrices throughout this section. A good overview of this line of acceleration techniques has been given by Smith, Ford, and Sidi (1987). See also Loisel and Takane (2009).

We first consider the case in which the updating equations are linear, and then adapt it to the nonlinear case. Let

$$\mathbf{x}^{(q+1)} = \mathbf{H}\mathbf{x}^{(q)} + \mathbf{b} \tag{25}$$

denote the updating equation, and let $\mathbf{x}_{conv.}$ represent a stationary point of (25). Then,

$$\mathbf{x}_{conv.} = (\mathbf{I} - \mathbf{H})^{-1}\mathbf{b}, \tag{26}$$

where it is assumed that $\mathbf{I} - \mathbf{H}$ is nonsingular. Let

$$\mathbf{u}^{(q)} = \mathbf{x}^{(q+1)} - \mathbf{x}^{(q)}, \tag{27}$$

and consider the sequence of vectors, $\mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \cdots, \mathbf{u}^{(k)}$, where the first k vectors are linearly independent, whereas the entire $k+1$ vectors are linearly dependent. That is,

$$\sum_{q=0}^{k} c_q \mathbf{u}^{(q)} = \mathbf{0} \tag{28}$$

for some c_q, $(q = 0, \cdots, k)$, where we further assume $c_k = 1$ (without loss of generality). Note that such a k always exists (for any set of vectors $\mathbf{u}^{(q)}$, $q = 0, \cdots, k$, if a sufficiently large k is chosen), and we tentatively assume that it is known in advance. Let $\mathbf{c} = [c_0, c_1, \cdots, c_{k-1}]'$, and $\mathbf{U} = [\mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \cdots, \mathbf{u}^{(k-1)}]$. Then,

$$\mathbf{U}\mathbf{c} = -\mathbf{u}^{(k)}, \tag{29}$$

so that

$$\mathbf{c} = -\mathbf{U}^{+}\mathbf{u}^{(k)}, \tag{30}$$

where $\mathbf{U}^{+}$ is the Moore-Penrose inverse of $\mathbf{U}$.

We note that $\mathbf{H}^q\mathbf{u}^{(0)} = \mathbf{u}^{(q)}$ in this particular case. Using this relation, we can rewrite (28) in the form of a polynomial equation in $\mathbf{H}$. That is,

$$\begin{aligned}
\mathbf{0} &= \sum_{q=0}^{k} c_q\mathbf{u}^{(q)} \\
&= c_0\mathbf{u}^{(0)} + c_1\mathbf{u}^{(1)} + c_2\mathbf{u}^{(2)} + \cdots, +c_k\mathbf{u}^{(k)}, \\
&= c_0\mathbf{u}^{(0)} + c_1\mathbf{H}\mathbf{u}^{(0)} + c_2\mathbf{H}^2\mathbf{u}^{(0)} + \cdots, +c_k\mathbf{H}^k\mathbf{u}^{(0)} \\
&= \sum_{q=0}^{k} c_q\mathbf{H}^q\mathbf{u}^{(0)} \\
&= \mathbf{P}(\mathbf{H})\mathbf{u}^{(0)}, \qquad (31)
\end{aligned}$$

where

$$\mathbf{P}(\mathbf{H}) = \sum_{j=1}^{k} c_j\mathbf{H}^j. \tag{32}$$

We say that $\mathbf{P}(\mathbf{H})$ is the minimal polynomial function of $\mathbf{H}$ that annihilates $\mathbf{u}^{(0)}$.

We also observe that

$$\mathbf{u}^{(0)} = (\mathbf{H} - \mathbf{I})(\mathbf{x}^{(0)} - \mathbf{x}_{conv.}), \tag{33}$$

where $\mathbf{x}_{conv.}$ is, as defined previously, the convergence point of $\mathbf{x}$, and that $\mathbf{H}-\mathbf{I}$ is invertible (see the remark made immediately after (26)) and commutes with $\mathbf{P}(\mathbf{H})$, so that

$$\mathbf{P}(\mathbf{H})(\mathbf{x}^{(0)} - \mathbf{x}_{conv.}) = \mathbf{0} \tag{34}$$

also holds. That is, the minimal polynomial $\mathbf{P}(\mathbf{H})$ that annihilates $\mathbf{u}^{(0)}$ also annihilates $\mathbf{x}^{(0)} - \mathbf{x}_{conv.}$. Hence, (31) can be further rewritten as

$$\begin{aligned}
\mathbf{0} &= \mathbf{P}(\mathbf{H})(\mathbf{x}^{(0)} - \mathbf{x}_{conv.}) \\
&= \sum_{q=0}^{k} c_q\mathbf{H}^q(\mathbf{x}^{(0)} - \mathbf{x}_{conv.}) \\
&= \sum_{q=0}^{k} c_q(\mathbf{x}^{(q)} - \mathbf{x}_{conv.}), \qquad (35)
\end{aligned}$$

(because $\mathbf{x}^{(j)} - \mathbf{x}_{conv.} = \mathbf{H}^q(\mathbf{x}^{(0)} - \mathbf{x}_{conv.})$). From (35), we obtain

$$\mathbf{x}_{conv.} = \sum_{q=0}^{k} c_q \mathbf{x}^{(q)} / \sum_{q=0}^{k} c_q. \tag{36}$$

That is, $\mathbf{x}_{conv.}$ gives an exact solution (convergence) point of $\mathbf{x}$, provided that (1) the basic iterate is linear, and that (2) the exact value of k is known.

However, the basic iteration formula in Takane's algorithm is not linear. Thus, the above procedure has to be applied repeatedly. The value of k is also not known in advance. Fortunately, however, this value does not have to be determined precisely. As far as the above procedure has to be applied repeatedly (because of the nonlinear updating equations), k can be any value for which the relation (28) holds only approximately. In the following numerical experiment, several values of k (= 1, 5, 10, 15, and 20) were tried to see which value gives the best performance.

Table 2 compares the performance of the MPE method with Takane's original algorithm. Again, the same data sets that were generated to obtain Table 1 were used in this study. Entries of the table indicate the mean cpu time for convergence over 100 replications. In the table, n indicates the size of the data matrix, r the number of components, and k the degree of polynomials in the MPE method. ($k = 1$ corresponds with Takane's original algorithm.) It can be seen that the MPE method achieves a substantial reduction in cpu time, particularly when the number of DEDICOM parameters to be estimated is large. An optimal value of k is around 15 in the present case, although obviously a more elaborate study is in order on this point.

***** Insert Table 2 about here. *****

Figure 1 shows a typical iteration history for Takane's original algorithm and the MPE method with $k = 10$. The top panel shows the value of the fit criterion (12) against iteration numbers, and the bottom panel shows the log of the norm of the projected gradients (the log of (20)) against iteration numbers. In the figure, the results from Takane's original method are labeled by "x", whereas those from the MPE method by "o". Note that both the fit value and the log norm of the projected gradient are obtained for every iteration for Takane's original algorithm ($k = 1$), while they were obtained only for every 10 iterations for the MPE method. For both methods a near optimal value of the fit criterion is attained rather quickly. However, the norm of the projected gradient goes down more quickly using the MPE method than for Takane's algorithm. This means that the MPE method is more suitable to obtain precise solutions. This degree of precision is often necessary because the fit criterion may be rather flat near the optimal point, which causes premature stopping unless a strict convergence criterion is used. The case of non-monotonic convergence has been found to be rather rare with the data generation scheme adopted here, particularly with Takane's algorithm, and even when it happens the amount of decrease in the fit value is in most cases so slight that it is difficult to see in a graph.

***** Insert Figure 1 about here *****

To investigate the seriousness of convergence to suboptimal solutions in the two algorithms (Takane's algorithm and the MPE method with $k = 10$), analyses were repeated with 15 different random initial estimates of $\mathbf{X}$. Neither of the two algorithms have shown clear superiority in avoiding suboptimal solutions. Overall, the chance of convergence to suboptimal solutions is remarkably low for both methods (about 5%), which are mostly concentrated in a few specific data sets. Out of 300 data sets, each analyzed under two different values of r (the number of components), there was no single instance in which the two methods yielded different best solutions among the 15 solutions they each obtained, a majority of which coincided with the best solutions. Note that these are the results from randomly generated data sets with no obvious DEDICOM structures. It is expected that the chance of suboptimal solutions is much less with real data sets with some built-in DEDICOM structures.

7 Concluding remarks

In this paper, we proposed some acceleration techniques for Takane's algorithm for DEDICOM, and demonstrated their usefulness numerically. In doing so, we made a "bold" proposition to ignore the monotonic convergence property with respect to the criterion function they optimized ((7) and (12)). Note, however, that at present there is no rigorous convergence proof of the algorithms without the monotonicity property. Nonetheless, it seemed worthwhile to explore what we could gain if we lifted the principle of monotonicity, and we were successful in demonstrating the positive effects of two simple acceleration techniques.

Takane's original algorithm is fairly efficient already for moderate sized problems, particularly when the data matrix is nearly EP$_r$. One might argue that there is little need to further accelerate the algorithm. There are situations, however, in which DEDICOM solutions have to be obtained repeatedly. For example, permutation tests for determining the number of significant components in the DEDICOM model would require repeated applications of the DEDICOM algorithm to matrices having no obvious DEDICOM structures. There is also a problem of non-global optima, as pointed out earlier. The only possible solution to the problem is to obtain solutions for the same data set starting from many different initial estimates. The relative infrequency of the occurrence of suboptimal solutions in the DEDICOM model is no consolation. As far as they exist, we have to avoid them, and this is again where a more efficient algorithm is called for.

References

Bauer, F.L. (1957). Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme. *ZAMP, 8*, 214-235.

Edelman, A., Arias, T., and Smith, S.T. (1997). The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Applications, 20*, 303-353.

Gower, J.C. (1977). The analysis of asymmetry and orthogonality. In J.R. Barra, F. Brodeau, G. Romier, B., and van Cutem (Eds.), *Recent developments in statistics*, (pp. 190-203). Amsterdam: North Holland.

Harshman, R.A. (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, Ontario.

Harshman, R.A., Green, P.E., Wind, Y., and Lundy, M.E. (1981). A model for the analysis of asymmetric data in marketing research. *Marketing Science, 1*, 205-242.

Jennrich, R.I. (2001). A simple general procedure for orthogonal rotation. *Psychometrika, 66*, 289-306.

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., Takane, Y., and de Leeuw, J. (1990). A generalization of Takane's Algorithm for DEDICOM. *Psychometrika, 55*, 151-158.

Loisel, S., and Takane, M. (2009). Fast indirect robust generalized method of moments. Submitted for publication to *Computational Statistics and Data Analysis.*

Ramsay, J.O. (1975). Solving implicit equations in psychometric data analysis. *Psychometrika, 40*, 337-360.

Rutishauser, H. (1969). Computational aspects of F. L. Bauer's simultaneous iteration method. *Numerische Mathematik, 13*, 4-13.

Smith, D.A., Ford, W.F., and Sidi, A. (1987). Extrapolation methods for vector sequences. *SIAM Review, 29*, 199-233.

Schwerdtfeger, H. (1950). *Introduction to linear algebra and the theory of matrices.* Groningen: P. Noordhoff.

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

Takane, Y. (2004). Matrices with special reference to applications in psychometrics. *Linear Algebra and Its Applications, 388C*, 341-361.

Tian, Y. (2008). EP and weighted EP matrices, Unpublished manuscript, McGill University.

Table 1: The mean cpu time with Takane's algorithm with orthonormalization in every several iterations.

		Orthnormalization in every		
n	r	1 iteration	5 iterations	10 iterations
10	3	0.0133	0.0117	0.0107
10	5	0.0147	0.0116	0.0116
50	5	0.1362	0.1006	0.0980
50	10	0.2804	0.2450	0.2402
100	5	0.6253	0.3209	0.3148
100	10	1.4424	0.7995	0.8000

Table 2: The mean cpu time with the MPE method as compared to Takane's original algorithm.

n	r	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
10	3	0.0133	0.0080	0.0070	0.0041	0.0042
10	5	0.0147	0.0124	0.0058	0.0052	0.0048
50	5	0.1362	0.0536	0.0491	0.0304	0.0480
50	10	0.2804	0.0901	0.0701	0.0601	0.1063
100	5	0.6253	0.2698	0.2597	0.2331	0.2329
100	10	1.4424	0.3500	0.3253	0.2254	0.3892

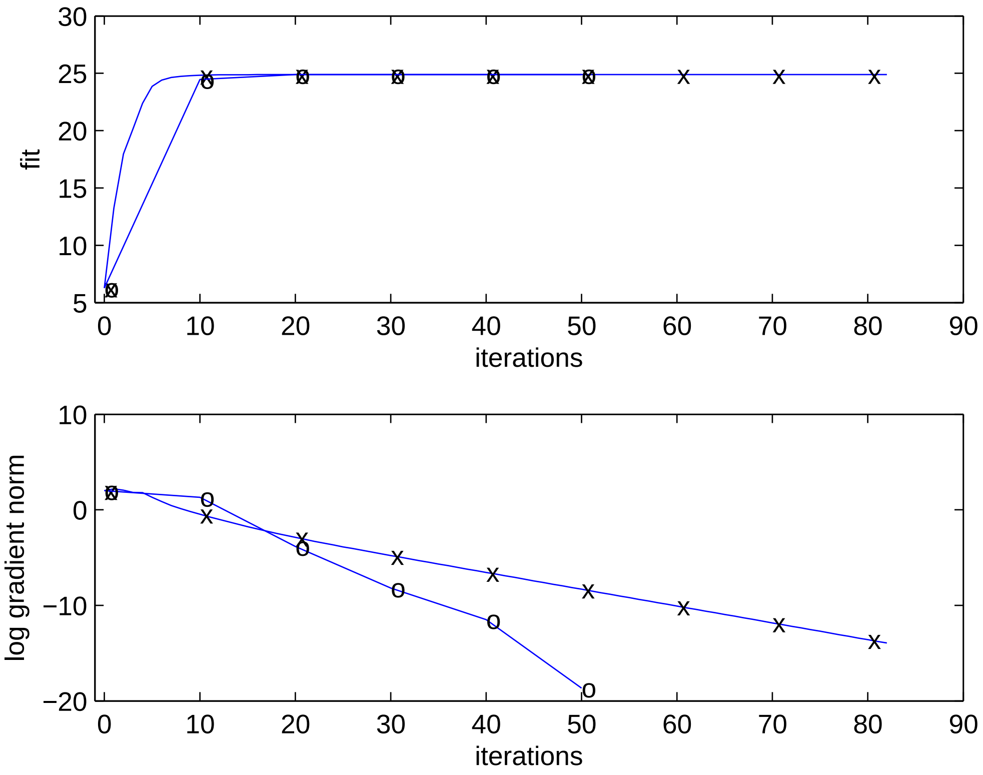

Figure 1: The fit index and the log norm of the projected gradients as a function of typical iterations. (x indicates no acceleration, and o indicates acceleration by the MPE method with $k = 10$.)