Weighted Majorization Algorithms for Weighted Least Squares Decomposition Models

Patrick J.F. Groenen* Patrizia Giaquinto †

Henk A. L. Kiers‡

March 10, 2003

Econometric Institute Report EI 2003-09

*Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands (e-mail: groenen@few.eur.nl). We would like to thank Anton Béguin and Norman Verhelst for their valuable remarks that have helped to improve this paper.

†Department of Statistical Sciences, University of Bari

‡Department of Psychology, University of Groningen

Abstract

For many least-squares decomposition models efficient algorithms are well known. A more difficult problem arises in decomposition models where each residual is weighted by a nonnegative value. A special case is principal components analysis with missing data. Kiers (1997) discusses an algorithm for minimizing weighted decomposition models by iterative majorization. In this paper, we propose a more efficient algorithm called weighted majorization for computing a solution. We will show that the algorithm by Kiers is a special case of our algorithm. Here, we will apply weighted majorization to weighted principal components analysis, robust Procrustes analysis, and logistic bi-additive models of which the two parameter logistic model in item response theory is a special case. Simulation studies show that weighted majorization is generally faster than the method by Kiers by a factor one to four and obtains the same or better quality solutions. For logistic bi-additive models, we propose a new iterative majorization algorithm called logistic majorization.

Keywords: Weighted principal component analysis, iterative majorization, robust Procrustes analysis, logistic bi-additive model, IRT, two parameter logistic model.

1 Introduction

Many least-squares decomposition problems have standard solutions that can be easily computed. A more difficult problem arises when nonnegative weights are attached to each residual. Such a situation comes up in iterated weighted least squares problems as a part in robust statistics and in generalized bi-additive modelling. Kiers (1997) discusses an iterative majorization algorithm that transforms the weighted least-squares problem in an unweighted least-squares problem. Especially in the case where one weight is much larger than the others, the algorithm by Kiers (1997) may become slow. Even though computers become faster over the time, accelerations are particularly welcome in applications to large data sets, such as in data mining, or in iterative algorithms. The main purpose of the current paper is to propose a weighted majorization algorithm and compare it to the one by Kiers (1997). We shall also show that Kiers' algorithm is a special case of ours. Therefore, our proposed algorithm is is expected to be faster than the one by Kiers.

To illustrate our central idea, we first need a least-squares decomposition model for which we take weighted principal components analysis (WPCA). The purpose of WPCA is to find a rank p subspace of the data that accounts for as much variance of the data as possible. Let $\mathbf{H}$ be an $n \times k$ matrix of data values, $\mathbf{X}$ an $n \times p$ matrix of object scores, and $\mathbf{A}$ a $k \times p$ matrix of component loadings. Then unweighted PCA can be defined as the minimization of

$$L_{\text{PCA}}(\mathbf{X}, \mathbf{A}) = \|\mathbf{H} - \mathbf{X}\mathbf{A}'\|^2, \tag{1}$$

where $\|\mathbf{B}\|^2 = \text{tr } \mathbf{B}'\mathbf{B} = \sum_{i=1}^{n}\sum_{j=1}^{k} b_{ij}^2$ denotes the usual sum of squares of $\mathbf{B}$. Then (1) can also be expressed as

$$L_{\text{PCA}}(\mathbf{X}, \mathbf{A}) \quad = \quad \sum_{i=1}^{n}\sum_{j=1}^{k}(h_{ij} - \mathbf{x}_i'\mathbf{a}_j)^2 = \sum_{i=1}^{n}\sum_{j=1}^{k} e_{ij}^2, \tag{2}$$

where $\mathbf{x}_i$ and $\mathbf{a}_j$ are column vectors of length p containing respectively the elements of row i of $\mathbf{X}$ and of row j of $\mathbf{A}$. A straightforward extension is to define weighted PCA (WPCA) as

$$L_{\text{WPCA}}(\mathbf{X}, \mathbf{A}) \quad = \quad \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}(h_{ij} - \mathbf{x}_i'\mathbf{a}_j)^2 = \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}e_{ij}^2 \tag{3}$$

for some given $n \times k$ matrix $\mathbf{W}$ of nonnegative weights w_{ij}. The weights can be used to indicate missing values ($w_{ij} = 1$ indicates nonmissing value, $w_{ij} = 0$ indicates a missing value) or attach value to indicate the importance to the data value h_{ij}. Thus, (3) can be seen as a weighted least-squares problem.

Kiers (1997) discussed an iterative majorization algorithm to minimize a weighted least-squares problem. The basic idea behind iterative majorization is that in each iteration, the original function $L(\mathbf{X})$ is substituted by an auxiliary function $\mu(\mathbf{X}, \mathbf{X}^0)$, the so called majorizing function, that meets several requirements. First, the majorizing function should touch the original function at the current estimate $\mathbf{X}^0$, that is, $L(\mathbf{X}^0) = \mu(\mathbf{X}^0, \mathbf{X}^0)$. Second, the majorizing function should be larger than or equal to the original function everywhere, that is, $L(\mathbf{X}) \leq \mu(\mathbf{X}, \mathbf{X}^0)$. Usually, the majorizing function is chosen to be simple, that is, linear or quadratic. Suppose that we can obtain the minimum $\mathbf{X}^+$ of the majorizing function $\mu(\mathbf{X}, \mathbf{X}^0)$. Then, the following chain of inequalities holds

$$L(\mathbf{X}^+) \leq \mu(\mathbf{X}^+, \mathbf{X}^0) \leq \mu(\mathbf{X}^0, \mathbf{X}^0) = L(\mathbf{X}^0). \tag{4}$$

This chain of inequalities guarantees that a series of nonincreasing loss function values is obtained. If $L(\mathbf{X})$ is bounded below or is sufficiently constrained, it can be proved that the iterations will stop at a stationary $\mathbf{X}$ that is not necessarily at a local minimum. However, in all practical cases $L(\mathbf{X})$ is at a local minimum after the function values have converged. For more information on iterative majorization, we refer to De Leeuw (1993), De Leeuw (1994), Heiser (1995), Borg and Groenen (1997), and Kiers (2002, in press).

The application of iterative majorization by Kiers (1997) is based on the following idea. Let $\mathbf{e}$ denote a column vector of the nk residuals and let $\mathbf{D}_{\mathbf{W}}$ be an $nk \times nk$ diagonal matrix with elements w_{ij}. Then (3) can be expressed as

$$L_{\text{WPCA}}(\mathbf{X}, \mathbf{A}) = \mathbf{e}'\mathbf{D}_{\mathbf{W}}\mathbf{e}. \tag{5}$$

Kiers (1997) applied a majorizing function proposed by Heiser (1987) and developed a quadratic majorizing function from the inequality

$$(\mathbf{e} - \mathbf{e}^0)'(\mathbf{D}_{\mathbf{W}} - m\mathbf{I})(\mathbf{e} - \mathbf{e}^0) \leq 0, \tag{6}$$

where $\mathbf{e}^0$ is the vector of residuals from the previous iteration and $m = w_{\max}$ is the maximum value of $\mathbf{D_W}$. Note that $\mathbf{e}^0$ contains estimates of $\mathbf{X}$ and $\mathbf{A}$ that are currently known. Inequality (6) holds because the matrix $\mathbf{D_W} - m\mathbf{I}$ is negative semi-definite, so that, $\mathbf{e}'(\mathbf{D_W} - m\mathbf{I})\mathbf{e} \leq 0$ for any vector $\mathbf{e}$. As required for iterative majorization, (6) becomes an equality for $\mathbf{e} = \mathbf{e}^0$. Expanding (6) and rewriting gives

$$\begin{aligned}
(\mathbf{e}-\mathbf{e}^0)'(\mathbf{D_W} - m\mathbf{I})(\mathbf{e}-\mathbf{e}^0) &\leq 0 \\
(\mathbf{e}-\mathbf{e}^0)'\mathbf{D_W}(\mathbf{e}-\mathbf{e}^0) &\leq m(\mathbf{e}-\mathbf{e}^0)'(\mathbf{e}-\mathbf{e}^0) \\
\mathbf{e}'\mathbf{D_W}\mathbf{e} + \mathbf{e}^{0'}\mathbf{D_W}\mathbf{e}^0 - 2\mathbf{e}'\mathbf{D_W}\mathbf{e}^0 &\leq m\mathbf{e}'\mathbf{e} + m\mathbf{e}^{0'}\mathbf{e}^0 - 2m\mathbf{e}'\mathbf{e}^0 \\
\mathbf{e}'\mathbf{D_W}\mathbf{e} &\leq m\mathbf{e}'\mathbf{e} - 2m\mathbf{e}'\mathbf{e}^0 + 2\mathbf{e}'\mathbf{D_W}\mathbf{e}^0 \\
&\quad -\mathbf{e}^{0'}\mathbf{D_W}\mathbf{e}^0 + m\mathbf{e}^{0'}\mathbf{e}^0 \\
\mathbf{e}'\mathbf{D_W}\mathbf{e} &\leq m\mathbf{e}'\mathbf{e} - 2m\mathbf{e}'[\mathbf{e}^0 - m^{-1}\mathbf{D_W}\mathbf{e}^0] \\
&\quad -\mathbf{e}^{0'}\mathbf{D_W}\mathbf{e}^0 + m\mathbf{e}^{0'}\mathbf{e}^0. \qquad (7)
\end{aligned}$$

Let $c_1 = m\mathbf{e}^{0'}\mathbf{e}^0 - \mathbf{e}^{0'}\mathbf{D_W}\mathbf{e}^0$ and $\mathbf{z} = \mathbf{e}^0 - m^{-1}\mathbf{D_W}\mathbf{e}^0$. Then we may express (7) as

$$\begin{aligned}
L_{\mathrm{WPCA}}(\mathbf{X},\mathbf{A}) = \mathbf{e}'\mathbf{D_W}\mathbf{e} &\leq m(\mathbf{e}-\mathbf{z})'(\mathbf{e}-\mathbf{z}) - m\mathbf{z}'\mathbf{z} + c_1 \\
&= \mu_1(\mathbf{X},\mathbf{A},\mathbf{X}^0,\mathbf{A}^0). \qquad (8)
\end{aligned}$$

From both (7) and (8) it can be seen that the weighted least squares function in the residuals $L_{\mathrm{WPCA}}(\mathbf{X},\mathbf{A})$ can be majorized by an *unweighted* least squares function in the residuals. For the latter function, many standard numerical methods are available to obtain an analytic solution. Therefore, we consider the approach by Kiers (1997) to be a very powerful one that has a wide applicability.

One of the disadvantages of the algorithm by Kiers (1997) can be seen if the majorizing function at the right hand side of (8) is examined more closely. The only important part is $(\mathbf{e}-\mathbf{z})'(\mathbf{e}-\mathbf{z})$, since the rest is constant. Note that $\mathbf{z}_i$ has elements

$$z_{ij} = e^0_{ij} - \frac{w_{ij}}{m}e^0_{ij}. \qquad (9)$$

We can write

$$\begin{aligned}
(\mathbf{e}-\mathbf{z})'(\mathbf{e}-\mathbf{z}) &= \sum_{i=1}^{n}\sum_{j=1}^{k}(e_{ij}-z_{ij})^2 = \sum_{i=1}^{n}\sum_{j=1}^{k}\left(e_{ij} - e^0_{ij} + \frac{w_{ij}}{m}e^0_{ij}\right)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}\left(h_{ij} - \mathbf{x}_i'\mathbf{a}_j - h_{ij} + \mathbf{x}_i^{0'}\mathbf{a}^0_j + \frac{w_{ij}}{m}(h_{ij} - \mathbf{x}_i^{0'}\mathbf{a}^0_j)\right)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}\left(\left[1 - \frac{w_{ij}}{m}\right]\mathbf{x}_i^{0'}\mathbf{a}^0_j + \frac{w_{ij}}{m}h_{ij} - \mathbf{x}_i'\mathbf{a}_j\right)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}(r_{ij} - \mathbf{x}_i'\mathbf{a}_j)^2 = \mathrm{tr}\ (\mathbf{R} - \mathbf{X}\mathbf{A}')'(\mathbf{R} - \mathbf{X}\mathbf{A}'), \qquad (10)
\end{aligned}$$

where

$$r_{ij} = \left[1 - \frac{w_{ij}}{m}\right] \mathbf{x}_i^{0\prime} \mathbf{a}_j^0 + \frac{w_{ij}}{m} h_{ij}. \tag{11}$$

Several things can be said about (10). First, if all weights w_{ij} are equal, then (10) simplifies into $\sum_{i=1}^{n} \sum_{j=1}^{k} (h_{ij} - \mathbf{x}_i' \mathbf{a}_j)^2$ because $w_{ij}/m = 1$ for all i, j. In this case, the majorizing function has simplified into the ordinary PCA model (2) that can be solved by a singular value decomposition in one step. The second observation touches one of the main topics of the current paper, that is the slowness of the algorithm. In (10) it can be seen that for all i, j we fit $\mathbf{x}_i' \mathbf{a}_j$ to the convex combination $[1 - w_{ij}/m]\mathbf{x}_i^{0\prime} \mathbf{a}_j^0 + (w_{ij}/m)h_{ij}$. Because $0 \leq w_{ij} \leq m$, we must have $0 \leq w_{ij}/m \leq 1$. Now, if there is a single weight w_{ij} that is much larger than all the other weights, then for most combinations i, j we have that $w_{ij}/m \ll 1$ so that $\mathbf{x}_i^{0\prime} \mathbf{a}_j^0$ dominates h_{ij}. In other words, when updating the majorizing function, the $\mathbf{x}_i' \mathbf{a}_j$ is fitted mostly to $\mathbf{x}_i^{0\prime} \mathbf{a}_j^0$ and only to a minor extent to h_{ij}. The consequence is that the more deviant the largest w_{ij} is from the other weights, the slower the algorithm will be. Slowness may especially be a problem in large data sets. In this paper, we propose an iterative majorization algorithm that is faster than the algorithm by Kiers (1997), and has his algorithm as a special case.

The remainder of this paper is organized as follows. In the next section, we discuss a weighted majorization algorithm that is faster or equally fast as Kiers' algorithm. The introduction of the algorithm is illustrated by weighted principal component analysis. Then, we apply the weighted algorithm to robust Procrustes analysis, and logistic bi-additive modelling. For each application of our weighted majorizing algorithm, we present a simulation study that either compares the speed of the algorithms or their quality. We end this paper with some conclusions and a discussion.

2 A Weighted Iterative Majorization Algorithm using Weighted Least Squares

In the previous section, we indicated that the larger the difference between the largest value of $\mathbf{W}$ and the remaining weights, the slower the algorithm will be. To improve the speed of the algorithm by Kiers (1997), it is important to realize that the loss function is a sum of the loss per row. The basic idea of our improvement is to develop a majorizing function for each row separately. We shall see that minimizing this majorizing function becomes a *weighted* least-squares problem in a diagonal metric instead of an unweighted least-squares. The advantage is that the weighted majorizing function is generally closer to the original function than the majorizing function in (8). For quite a few least-squares problems, imposing a diagonal weight matrix does not make it more difficult to solve. Without loss of generality, it is assumed throughout this paper that the number of rows n is equal to or larger than the number of columns k.

To develop the weighted majorizing function, the methodology from the previous section is applied in a row by row fashion. Let $\mathbf{D}_{\mathbf{w}_i}$ a $k \times k$ matrix of the vector of weights $\mathbf{w}_i$ corresponding to row i. Then L_{WPCA} can be expressed as

$$L_{\text{WPCA}}(\mathbf{X}, \mathbf{A}) = \mathbf{e}'\mathbf{D}_{\mathbf{W}}\mathbf{e} = \sum_{i=1}^{n} \mathbf{e}_i'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i, \tag{12}$$

where $\mathbf{e}_i$ is the column vector of errors corresponding to row i. To apply the majorizing inequality (7) to a single row, define $\mathbf{m}$ as an n vector containing the maximum row values of $\mathbf{W}$, so that $\mathbf{m}$ has elements $m_i = \max_j w_{ij}$. Adding a subscript for row i to (6) allows us to develop a majorizing inequality for each row, that is,

$$(\mathbf{e}_i - \mathbf{e}_i^0)'(\mathbf{D}_{\mathbf{w}_i} - m_i\mathbf{I})(\mathbf{e}_i - \mathbf{e}_i^0) \leq 0,$$

which holds since $\mathbf{D}_{\mathbf{w}_i} - m_i\mathbf{I}$ is negative semi-definite. Further expansions show that

$$\mathbf{e}_i'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i + \mathbf{e}_i^{0\prime}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 - 2\mathbf{e}_i{}'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 - m_i\mathbf{e}_i{}'\mathbf{e}_i - m_i\mathbf{e}_i^{0\prime}\mathbf{e}_i^0 + 2m_i\mathbf{e}_i{}'\mathbf{e}_i^0 \leq 0.$$

Rearranging gives the row wise majorizing inequality

$$\mathbf{e}_i'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i \leq m_i\mathbf{e}_i{}'\mathbf{e}_i - 2m_i\mathbf{e}_i{}'\mathbf{e}_i^0 + 2\mathbf{e}_i{}'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 + c_{2i}, \tag{13}$$

where $c_{2i} = m_i\mathbf{e}_i^{0\prime}\mathbf{e}_i^0 - \mathbf{e}_i^{0\prime}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0$. Thus, instead of a single majorizing function for all error terms in the loss function, we have different majorizing functions for each of the n rows. Summing (13) over all the rows gives the weighted majorizing function at the right side of

$$\begin{aligned}
L_{\text{WPCA}}(\mathbf{X}, \mathbf{A}) &= \sum_{i=1}^{n} \mathbf{e}_i{}'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i \\
&\leq \sum_{i=1}^{n} m_i\mathbf{e}_i{}'\mathbf{e}_i - 2\sum_{i=1}^{n} m_i\mathbf{e}_i{}'\mathbf{e}_i^0 + 2\sum_{i=1}^{n} \mathbf{e}_i{}'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 + \sum_{i=1}^{n} c_{2i}. \\
&= \sum_{i=1}^{n} m_i\mathbf{e}_i{}'\mathbf{e}_i - 2\sum_{i=1}^{n} m_i\mathbf{e}_i{}'[\mathbf{e}_i{}^0 - m_i^{-1}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0] + \sum_{i=1}^{n} c_{2i} \\
&= \mu_2(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0).
\end{aligned} \tag{14}$$

Let $\mathbf{D}_{\mathbf{m}}$ be the diagonal matrix with $\mathbf{m}$ on its diagonal. Also, let $c_3 = \sum_{i=1}^{n} c_{2i}$ and row i of $\mathbf{Z}$ be $\mathbf{z}_i = \mathbf{e}_i^0 - m_i^{-1}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0$. Then we may express $\mu_2(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0)$ as

$$\begin{aligned}
\mu_2(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0) &= \sum_i (\mathbf{e}_i - \mathbf{z}_i)'\mathbf{D}_{\mathbf{m}}(\mathbf{e}_i - \mathbf{z}_i) - \sum_i m_i\mathbf{z}_i'\mathbf{z}_i + c_3 \\
&= \text{tr } (\mathbf{E} - \mathbf{Z})'\mathbf{D}_{\mathbf{m}}(\mathbf{E} - \mathbf{Z}) - \text{tr } \mathbf{Z}'\mathbf{D}_{\mathbf{m}}\mathbf{Z} + c_3,
\end{aligned} \tag{15}$$

where the elements of $\mathbf{z}_i$ are defined by

$$z_{ij} = e^0_{ij} - \frac{w_{ij}}{m_i} e^0_{ij}. \tag{16}$$

The main difference of the present method and the one by Kiers (1997) lies in the difference between (16) and (9), that is, the denominator m_i in (16) depends on the largest weight *per row*, whereas the denominator m in (9) depends on the *overall* largest weight.

Clearly, the only important part of (15) is tr $(\mathbf{E}-\mathbf{Z})'\mathbf{D_m}(\mathbf{E}-\mathbf{Z})$ that becomes

$$\begin{aligned}
&\text{tr } (\mathbf{E}-\mathbf{Z})'\mathbf{D_m}(\mathbf{E}-\mathbf{Z}) \\
&= \sum_{i=1}^{n} m_i \sum_{j=1}^{k} (e_{ij}-z_{ij})^2 = \sum_{i=1}^{n} m_i \sum_{j=1}^{k} \left(e_{ij} - e^0_{ij} + \frac{w_{ij}}{m_i} e^0_{ij} \right)^2 \\
&= \sum_{i=1}^{n} m_i \sum_{j=1}^{k} \left(h_{ij} - \mathbf{x}_i'\mathbf{a}_j - h_{ij} + \mathbf{x}_i^{0\prime}\mathbf{a}^0_j + \frac{w_{ij}}{m_i}(h_{ij} - \mathbf{x}_i^{0\prime}\mathbf{a}^0_j) \right)^2 \\
&= \sum_{i=1}^{n} m_i \sum_{j=1}^{k} \left(\left[1 - \frac{w_{ij}}{m_i} \right] \mathbf{x}_i^{0\prime}\mathbf{a}^0_j + \frac{w_{ij}}{m_i} h_{ij} - \mathbf{x}_i'\mathbf{a}_j \right)^2 \\
&= \sum_{i=1}^{n} m_i \sum_{j=1}^{k} (r_{ij} - \mathbf{x}_i'\mathbf{a}_j)^2 = \text{tr } (\mathbf{R}-\mathbf{X}\mathbf{A}')'\mathbf{D_m}(\mathbf{R}-\mathbf{X}\mathbf{A}')
\end{aligned} \tag{17}$$

with

$$r_{ij} = \left[1 - \frac{w_{ij}}{m_i} \right] \mathbf{x}_i^{0\prime}\mathbf{a}^0_j + \frac{w_{ij}}{m_i} h_{ij}. \tag{18}$$

It can be seen in (18) that r_{ij} is still a convex combination of the previous estimates $\mathbf{x}_i^{0\prime}\mathbf{a}^0_j$ and the data h_{ij}. The main advantage of (17) above (10) is that each row is majorized separately. This property means that influence of a large weight is limited to its own row. For this row, $\mathbf{x}_i'\mathbf{a}_j$ is mostly fitted to $\mathbf{x}_i^{0\prime}\mathbf{a}^0_j$ and to minor extent to h_{ij}. The other rows remain unaffected. In rows that have their weights close to the largest row weight, the $\mathbf{x}_i'\mathbf{a}_j$ are mostly fitted to the data h_{ij} and only partially to the previous estimates $\mathbf{x}_i^{0\prime}\mathbf{a}^0_j$.

The final step in obtaining an update of the weighted majorizing function $\mu_2(\mathbf{X},\mathbf{A},\mathbf{X}^0,\mathbf{A}^0)$ is minimizing tr $(\mathbf{R}-\mathbf{X}\mathbf{A}')'\mathbf{D_m}(\mathbf{R}-\mathbf{X}\mathbf{A}')$ with respect to $\mathbf{X}$ and $\mathbf{A}$. This row-weighted loss function may be written as

$$\text{tr } (\mathbf{R}-\mathbf{X}\mathbf{A}')'\mathbf{D_m}(\mathbf{R}-\mathbf{X}\mathbf{A}') = \text{tr } (\mathbf{D}_\mathbf{m}^{1/2}\mathbf{R} - \mathbf{D}_\mathbf{m}^{1/2}\mathbf{X}\mathbf{A}')'(\mathbf{D}_\mathbf{m}^{1/2}\mathbf{R} - \mathbf{D}_\mathbf{m}^{1/2}\mathbf{X}\mathbf{A}'). \tag{19}$$

It is well known that the right hand part of (19) is minimized by computing the singular value decomposition $\mathbf{K}\mathbf{\Lambda}\mathbf{L}'$ of $\mathbf{D}_\mathbf{m}^{1/2}\mathbf{R}$, where $\mathbf{K}$ and $\mathbf{L}$ are orthonormal and $\mathbf{\Lambda}$ is diagonal with nonnegative elements. The least squares solution of (19) is obtained by setting

$$\mathbf{X} = \mathbf{D}_\mathbf{m}^{-1/2}\mathbf{K}_p \text{ and } \mathbf{A} = \mathbf{Q}_p\mathbf{\Lambda}_p, \tag{20}$$

where $\mathbf{K}_p, \mathbf{Q}_p$, and $\mathbf{\Lambda}_p$ denote the first p columns (and rows for $\mathbf{\Lambda}_p$) of $\mathbf{K}, \mathbf{L}$, and $\mathbf{\Lambda}$.

The weighted majorization algorithm for WPCA can be summarized as follows:

1. Initialize $\mathbf{X}_0$ and $\mathbf{A}_0$. Set iteration counter $l = 0$. Set convergence criterion ϵ to some small value.

2. $l := l + 1$.

3. Compute $\mathbf{R}$ according to (18) where $\mathbf{X}^0 = \mathbf{X}_{l-1}$ and $\mathbf{A}^0 = \mathbf{A}_{l-1}$.

4. Compute $\mathbf{X}_l$ and $\mathbf{A}_l$ according to(20).

5. If $L_{\text{WPCA}}(\mathbf{X}_{l-1}, \mathbf{A}_{l-1}) - L_{\text{WPCA}}(\mathbf{X}_l, \mathbf{A}_l) > (\sum_{ij} w_{ij} h_{ij}^2)\epsilon$ then go to Step 2.

6. Consider the algorithm converged.

The algorithm by Kiers (1997) differs from the weighted majorization algorithm in the following aspects. First, in Step 3, $\mathbf{R}$ is not computed by (18) but by (11). Clearly, if $m_i = m$ for all i, then (18) and (11) are equivalent. Second, Step 4 of Kiers' algorithm is different to Step 4 of the weighted algorithm in that $\mathbf{D_m}$ is chosen as $m\mathbf{I}$ by Kiers. Except for these two differences, both algorithms are the same.

Some properties of our weighted algorithm can be derived. Firstly, the algorithm by Kiers (1997) coincides with our algorithm when the largest value of w_{ij} per row is the same for all rows, that is, $m_i = m$ for all i. This situation includes the important case of missing data imputation where w_{ij} is either zero or one. For these cases, no improvement in computational efficiency can be expected with our method. For unequal maximum row values, our method will generally be faster. We expect the largest difference in efficiency whenever there are just a few large weights and the remaining weights are more or less of the same size. It is difficult to prove any results on how much faster our algorithm will be. Therefore, we revert to a comparison in simulation studies in subsequent sections.

Another property of weighted majorization is that its majorizing function is always smaller than or equal to the one by Kiers (1997), that is,

$$L(\mathbf{X}, \mathbf{A}) \leq \mu_2(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0) \leq \mu_1(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0). \tag{21}$$

This property can be proven as follows. Remember that m_i is the largest weight of row i and that m is the overall largest weight. Then, we must have $m_i \leq m$ or equivalently $-m \leq -m_i$ for all i, and thus

$$(\mathbf{e}_i - \mathbf{e}_i^0)'(\mathbf{D}_{\mathbf{w}_i} - m\mathbf{I})(\mathbf{e}_i - \mathbf{e}_i^0) \leq (\mathbf{e}_i - \mathbf{e}_i^0)'(\mathbf{D}_{\mathbf{w}_i} - m_i\mathbf{I})(\mathbf{e}_i - \mathbf{e}_i^0) \leq 0.$$

Summing over i and rearranging gives

$$\sum_i \mathbf{e}_i \mathbf{D}_{\mathbf{w}_i} \mathbf{e}_i - \sum_i \left[m\mathbf{e}_i{}'\mathbf{e}_i + m\mathbf{e}_i^{0\prime}\mathbf{e}_i^0 - 2m\mathbf{e}_i{}'\mathbf{e}_i^0 + 2\mathbf{e}_i{}'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 - \mathbf{e}_i^{0\prime}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 \right] \leq$$
$$\sum_i \mathbf{e}_i \mathbf{D}_{\mathbf{w}_i} \mathbf{e}_i - \sum_i \left[m_i\mathbf{e}_i{}'\mathbf{e}_i + m_i\mathbf{e}_i^{0\prime}\mathbf{e}_i^0 - 2m_i\mathbf{e}_i{}'\mathbf{e}_i^0 + 2\mathbf{e}_i{}'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 - \mathbf{e}_i^{0\prime}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0 \right]. \quad (22)$$

Inequality (22) can be simplified into

$$L(\mathbf{X}, \mathbf{A}) - \mu_1(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0) \leq L(\mathbf{X}, \mathbf{A}) - \mu_2(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0).$$

Subtracting $L(\mathbf{X}, \mathbf{A})$ from both sides of the inequality and multiplying by -1 gives

$$\mu_2(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0) \leq \mu_1(\mathbf{X}, \mathbf{A}, \mathbf{X}^0, \mathbf{A}^0), \quad (23)$$

which proves that the weighted majorizing function is indeed always smaller than or equal to the majorizing function used by Kiers (1997). Combining (23) with the majorizing inequality (14) completes the proof of (21).

The use of the current algorithm is not limited to WPCA. The ideas of the algorithm can be applied to any weighted least-squares model with differential nonnegative weights for every residual. Of course, it makes only sense if the minimum for the row weighted quadratic majorizing function such as (17) can be easily obtained. The weighted algorithm will be more beneficial if the number of rows is much larger than the number of columns, because the majorizing function can stay closer to the original function. If the number of columns is much larger than the number of rows, then more efficiency is gained by applying majorization to each column instead of the row majorization discussed so far. Of course, the principle remains equivalent.

2.1 A Comparison Study

In the previous section, we discussed the basic ideas behind our weighted majorization algorithm. We expect our improved algorithm to be faster, but still have to study how much faster it is. For this purpose we have set up a simulation study to observe the performance of the two algorithms. We decided to include also the iterative criss-cross regression algorithm, proposed by Gabriel and Zamir (1979), as it performed well in the study by Kiers (1997).

The dependent variables in this study are the speed and the quality of the solutions. In the comparison of the algorithm by Kiers (1997) and our weighted algorithm, we consider the ratio of the number of iterations used by Kiers' algorithm to the number of iterations used by our weighted algorithm. Such a direct comparison is fair, as computationally about the same amount of operations is needed to be performed per iteration, according to (11) and (18). In the comparison of our weighted majorization algorithm and criss-cross regression, the ratio of CPU-time is used. The reason is that both algorithms are completely

different, so comparing numers of iterations would not be fair. All computations where done in MatLab 6.5 running under Windows NT.

We varied the following factors in our study: the number of rows (n = 50, 100, 200, 500), the number of columns (k = 10, 20, 40), the number of components (p = 2, 4, 8), and the type of weights using two levels (w = 0, 5).

For each combination of the factors, an $n \times k$ matrix $\mathbf{H}$ was generated randomly from the uniform $[0, 1]$ distribution. For level $w = 0$, $\mathbf{W}$ was also generated similar to $\mathbf{H}$ ensuring that all the weights are nonnegative as required. For level $w = 5$, the same procedure was used, except that randomly 5% of the elements $\mathbf{W}$ were multiplied by the factor five. This operation introduces a small number of large weights.

Each of the 72 combinations of factor levels was replicated five times, yielding a total number of 360 comparisons. For every comparison, the three algorithms were started by the same random $\mathbf{A}$ and $\mathbf{X}$. The convergence criterion ϵ was fixed at 10^{-8}, where the range of function values was between .03 and .89. In the comparisons, we only took into account those runs that ended (approximately) at the same local minimum, which was defined by both function values having at least four decimal places equal. In the comparison with Kiers' algorithm, this was the case in 231 runs (64%). For criss-cross regression, both algorithms led to the same local minimum in 156 runs (43%).

To see how the effects influence the speed, we did an analysis of variance (ANOVA) on the two ratio's (see Table 1). Instead of the ratio itself, we used $\log_{10}$ of the ratio to satisfy the assumption of homogeneity of the variances within the groups defined by the combinations of all factor levels. In the comparison of Kiers' algorithm with our algorithm, the biggest effects were found for the intercept and the main effect w, and these were also statistically significant at all common significance levels ($p < .001$). In the comparison of criss-cross and our algorithm, the effects for the intercept, the main effects w, n, and k were the biggest, these were also statistically significant at all common significance levels ($p < .001$, see Table 1).

To evaluate the interesting effects, the mean of $\log_{10}$ of the ratio was computed for each level of each significant effect. To facilitate interpretation, those values are transformed back to the scale of the ratio by using them as the exponent of 10, leading to a special kind of average of the ratios. The results are displayed in Table 2. It can be seen that our method is on 'average' about 2.4 times as fast as the one by Kiers. As the number of columns increases (thus k gets larger) the gain in speed reduces. If there is a small proportion of large weights, our method is about three times faster than the one by Kiers.

In the comparison of the speed of our algorithm to criss-cross regression, we have to interpret the ratio of CPU-time with some care as we cannot be sure that both methods are implemented equally efficient in MatLab. The results in Table 2 suggest that our method is overall on 'average' 1.8 times faster than criss-cross regression. As n gets larger, the gain in speed of our method is larger. For small k ($k = 10$) our method is on 'average' 3.3 times faster than criss-cross regression, but for $k = 40$ our method is slower than criss-cross regression by a factor .8. If there are some extreme weights and k is larger, then criss-cross

Table 1: The results of the analysis of variance (ANOVA) in a full factorial model for the log of the ratio of iterations in Kiers' algorithm (1997) and in the weighted majorization and for the log of the ratio of CPU in Criss-Cross regression and the weighted majorization. The test F is computed with the degrees of freedom displayed and those of the relative error (200 and 124 respectively).

	Comparison Kiers vs. weighted majorization log Ratio Iterations				Comparison Criss-Cross vs. weighted majorization log Ratio CPU			
Effect	df	F	p	η_p^2	df	F	p	η_p^2
Intercept	1	752.919	.000	.790	1	98.642	.000	.443
w	1	55.258	.000	.216	1	33.077	.000	.211
n	3	2.114	.100	.031	3	9.227	.000	.182
k	2	5.211	.006	.050	2	49.599	.000	.444
p	2	1.142	.321	.011	2	2.026	.136	.032
$w \times n$	3	.617	.605	.009	3	.322	.809	.008
$w \times k$	2	1.522	.221	.015	2	5.687	.004	.084
$w \times p$	2	.065	.937	.001	2	.119	.888	.002
$n \times k$	6	1.180	.319	.034	6	1.140	.343	.052
$n \times p$	6	1.003	.424	.029	6	1.173	.325	.054
$k \times p$	4	2.186	.072	.042	4	2.427	.051	.073
Error	200				124			
Total	231				156			

regression is two times faster than our method. Conversely, for no extreme weights and small k, our method is faster by a factor 3.6. These results suggest that our method is faster than criss-cross regression especially when k is small or n is large.

To compare the quality of the solutions, we simply counted how often our method obtained a higher loss, a lower loss, or approximately the same loss (that is, the absolute difference in loss of the two methods is less than 10^{-4}). Note that in this comparison the loss is divided by nk to make it independent of n and k. Our method obtained lower loss than the method by Kiers in 307 runs (85%), the same loss in 231 runs (64%), and higher loss in 53 runs (15%). These results indicate that our method is not only faster than the one by Kiers, but also yields the same or better quality solutions, although the difference in terms of loss seems to be small. Comparing the loss of our method to criss-cross regression, we found that in 131 runs (36%) our method obtained lower loss, in 156 runs (43%) the same loss, and in 229 runs (63%) a higher loss. Thus it seems that criss-cross regression often takes longer to compute a solution, but also tends to find a lower loss. Again, the differences in loss seem to be small.

3 Robust Procrustes Analysis

In this section, the weighted majorization algorithm is applied to robust Procrustes analysis. First, we discuss weighted Procrustes analysis and then we

Table 2: Significant effects in the ANOVA reported in Table 1. For the comparison of the algorithm by Kiers (1997) and the weighted majorization algorithm, the average ratio of iterations is reported. For the comparison of the criss-cross regression and the weighted majorization algorithm, the 'average' ratio of CPU-time is reported.

Effect					
Comparison Kiers and weighted majorization					
overall	2.4				
$w = 0$	1.9				
$w = 5$	3.0				
$k = 10$	2.6				
$k = 20$	2.5				
$k = 40$	2.1				
Comparison criss-cross and weighted majorization					
overall	1.8		$k = 10$	$k = 20$	$k = 40$
$w = 0$	2.4	$w = 0$	3.6	3.0	1.5
$w = 5$	1.3	$w = 5$	3.4	1.4	0.5
$k = 10$	3.3				
$k = 20$	2.1		$k = 10$	$k = 20$	$k = 40$
$k = 40$	0.8	$p = 2$	4.0	1.5	0.7
$n = 50$	1.2	$p = 4$	3.1	2.2	0.7
$n = 100$	1.5	$p = 8$	3.0	2.6	1.2
$n = 200$	2.0				
$n = 500$	2.8				

show how this problem comes up in the robust Procrustes analysis. Finally, we compare the quality of the solutions obtained by both algorithms.

In Procrustes analysis, we deal with the problem of fitting a model $\mathbf{XT}$ to a target data matrix $\mathbf{Y}$. Here, $\mathbf{X}$ and $\mathbf{Y}$ are $n \times k$ known configuration matrices, $\mathbf{W}$ is a matrix of weights of the same size and $\mathbf{T}$ is a $k \times k$ unknown rotation matrix. The orthogonal Procrustes rotation problem allows to change $\mathbf{X}$ through geometrical operations like rotation or reflection in order to match its target $\mathbf{Y}$ as best as possible. Here, we include differential nonnegative weights for each residual, which is a subproblem in robust Procrustes analysis, as we will see later in this section. For the moment, we stick to the weighted Procrustes problem defined as

$$L_{\text{WProc}}(\mathbf{T}) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}(y_{ij} - \mathbf{x}_i'\mathbf{t}_j)^2, \tag{24}$$

under the constraint that $\mathbf{T}$ is orthonormal, that is, $\mathbf{T}'\mathbf{T} = \mathbf{I}$. $L_{\text{WProc}}(\mathbf{T})$, too, can be seen as a weighted least-squares problem with special constraints. We can apply our method illustrated for L_{WPCA}, and consider (24) in a row by row fashion. Define the residual of element ij in the model as $\mathbf{e}_{ij} = y_{ij} - \mathbf{x}_i'\mathbf{t}_j$ and let $\mathbf{e}_{ij}^0 = y_{ij} - \mathbf{x}_i'\mathbf{t}_j^0$. Similarly to (12), loss function (24) can be expressed as

$$L_{\text{WProc}}(\mathbf{T}) = \mathbf{e}'\mathbf{D}_{\mathbf{W}}\mathbf{e} = \sum_{i=1}^{n} \mathbf{e}_i'\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i. \tag{25}$$

Using (15), $L_{\text{WProc}}(\mathbf{T})$ is majorized by the weighted function

$$\mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0) = \text{tr } (\mathbf{E} - \mathbf{Z})'\mathbf{D}_{\mathbf{m}}(\mathbf{E} - \mathbf{Z}) - \text{tr } \mathbf{Z}'\mathbf{D}_{\mathbf{m}}\mathbf{Z} + c_4,$$

with z_{ij} defined by (16) as before and $c_4 = \sum_{i=1}^{n}(m_i\mathbf{e}_i^{0\prime}\mathbf{e}_i^0 - \mathbf{e}_i^{0\prime}\mathbf{D}_{\mathbf{w}_i}\mathbf{e}_i^0)$. The first term can be expanded by

$$\begin{aligned} \text{tr } (\mathbf{E} - \mathbf{Z})'\mathbf{D}_{\mathbf{m}}(\mathbf{E} - \mathbf{Z}) &= \sum_{i=1}^{n} m_i \sum_{j=1}^{k}(r_{ij} - \mathbf{x}_i'\mathbf{t}_j)^2 \\ &= \text{tr } (\mathbf{R} - \mathbf{X}\mathbf{T}')'\mathbf{D}_{\mathbf{m}}(\mathbf{R} - \mathbf{X}\mathbf{T}'), \end{aligned} \tag{26}$$

where $\mathbf{R}$ is defined by

$$r_{ij} = \left[1 - \frac{w_{ij}}{m_i}\right]\mathbf{x}_i^{0\prime}\mathbf{t}_j^0 + \frac{w_{ij}}{m_i}y_{ij}. \tag{27}$$

Expanding (26) and taking the constraint $\mathbf{T}'\mathbf{T} = \mathbf{T}\mathbf{T}' = \mathbf{I}$ into account shows that

$$\begin{aligned} &\text{tr } (\mathbf{R} - \mathbf{X}\mathbf{T}')'\mathbf{D}_{\mathbf{m}}(\mathbf{R} - \mathbf{X}\mathbf{T}') \\ &= \text{tr } \mathbf{R}'\mathbf{D}_{\mathbf{m}}\mathbf{R} + \text{tr } \mathbf{T}\mathbf{X}'\mathbf{D}_{\mathbf{m}}\mathbf{X}\mathbf{T}' - 2\text{tr } \mathbf{T}'\mathbf{X}\mathbf{D}_{\mathbf{m}}\mathbf{R} \\ &= \text{tr } \mathbf{R}'\mathbf{D}_{\mathbf{m}}\mathbf{R} + \text{tr } \mathbf{X}'\mathbf{D}_{\mathbf{m}}\mathbf{X} - 2\text{tr } \mathbf{T}'\mathbf{X}\mathbf{D}_{\mathbf{m}}\mathbf{R} \\ &= c_5 - 2\text{tr } \mathbf{T}'\mathbf{X}\mathbf{D}_{\mathbf{m}}\mathbf{R}. \end{aligned}$$

Finally, we can majorize (24) by

$$L_{\text{WProc}}(\mathbf{T}) \leq c_4 + c_5 - 2\text{tr } \mathbf{T}'\mathbf{S} = \mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0), \tag{28}$$

where $\mathbf{S} = \mathbf{X}\mathbf{D}_\mathbf{m}\mathbf{R}$. Looking at (28) one can easily realize that minimizing $\mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0)$ over $\mathbf{T}$ is equivalent to maximizing the term tr $\mathbf{T}'\mathbf{S}$. Define the singular value decomposition (SVD) of $\mathbf{S}$ as $\mathbf{S} = \mathbf{K}\mathbf{\Lambda}\mathbf{L}'$ with $\mathbf{K}$ and $\mathbf{L}'$ orthonormal matrices (that is, $\mathbf{K}'\mathbf{K} = \mathbf{L}'\mathbf{L} = \mathbf{I}$) and $\mathbf{\Lambda}$ diagonal. According to Ten Berge (1993), the upper bound of tr $\mathbf{T}'\mathbf{S}$ is attained at $\mathbf{T} = \mathbf{K}\mathbf{L}'$.

Above we considered the weighted Procrustes problem. Below we show how this problem comes up in a robustified version of Procrustes analysis as proposed by Verboon and Heiser (1992), see also Verboon (1994). It may happen that $\mathbf{X}$ contains some outlying points giving rise to high residuals. In such situations, functions which try to curb the influence of those outliers are desirable. The general idea is to down weight large residuals in the loss function. Several possibilities for down weighting have been proposed in the literature, see, for example, Huber (1964), Beaton and Tukey (1974) and Mosteller and Tukey (1977). In this paper, we consider the absolute value of the residual as a robustifier, so that the loss function becomes

$$L_{\text{RProc}}(\mathbf{T}) = \sum_{i=1}^{n}\sum_{j=1}^{k} \mid y_{ij} - \mathbf{t}_i'\mathbf{x}_j \mid = \sum_{i=1}^{n}\sum_{j=1}^{k} \mid e_{ij} \mid \tag{29}$$

with e_{ij} defined just as before.

A majorizing function for (29) can be easily derived by applying the inequality

$$\mid e_{ij} \mid \leq \tfrac{1}{2}\frac{e_{ij}^2}{\mid e_{ij}^0 \mid} + \tfrac{1}{2} \mid e_{ij}^0 \mid, \tag{30}$$

see Heiser (1987). Note that when $e_{ij}^0 = 0$, then (30) is not defined. Therefore, as suggested by Heiser (1987) in such cases, we replace e_{ij}^0 by a small ϵ. The majorizing inequalities do not hold any more, but by making ϵ small enough, the chain of inequalities (4) turns out to be still valid in practice. Summing (30) over i and j gives:

$$L_{\text{RProc}}(\mathbf{T}) \leq \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{k} \frac{1}{\mid e_{ij}^0 \mid}\ e_{ij}^2 + \tfrac{1}{2}L_{\text{RProc}}(\mathbf{T}^0) = \mu_{\text{RProc}}(\mathbf{T}, \mathbf{T}^0). \tag{31}$$

If we define $w_{ij} = \mid e_{ij}^0 \mid^{-1}$, then the first term of $\mu_{\text{RProc}}(\mathbf{T}, \mathbf{T}^0)$ equals $L_{\text{WProc}}(\mathbf{T})$. This definition of the w_{ij}'s means that it is possible to combine (28) and (31) also in a different way as:

$$L_{\text{RProc}}(\mathbf{T}) \leq \tfrac{1}{2}L_{\text{WProc}}(\mathbf{T}) + \tfrac{1}{2}L_{\text{RProc}}(\mathbf{T}^0) \leq \tfrac{1}{2}\mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0) + \tfrac{1}{2}L_{\text{RProc}}(\mathbf{T}^0).$$

This expression shows that $L_{\text{RProc}}(\mathbf{T})$ can be majorized by a constant plus one half times the majorizing function of weighted Procrustes analysis $\mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0)$.

As noted below (28), a minimum of $\mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0)$ can be obtained in a single step.

The weighted majorization algorithm for robust Procrustes analysis is similar to the one for WPCA. Of course, $\mathbf{R}$ is computed by (27), the weights should be chosen as $w_{ij} =\mid e_{ij}^0 \mid^{-1}$, and the computation of the update for $\mu_{\text{WProc}}(\mathbf{T}, \mathbf{T}^0)$ is indicated above.

3.1 A Simulation Study

Above we described our weighted majorization algorithm used in the algorithm for the robust Procrustes problem, in which in each step the weighted least squares function L_{WProc} is decreased. An alternative procedure for the latter can be based on Kiers' (1997) algorithm for minimizing L_{WProc}. To see how well our method compares to the one using Kiers' (1997) procedure within the algorithm, we did a simulation study for the robust orthogonal Procrustes problem. In the algorithm, we stopped the iterations whenever the $L_{\text{WProc}}(\mathbf{T}^{l-1}) - L_{\text{WProc}}(\mathbf{T}^l)$ is smaller than $10^{-8} \sum_{ij} |y_{ij}|$.

The data were generated as follows. The data matrix $\mathbf{X}$ was drawn randomly from the standard normal distribution. Then, we created a random rotation matrix $\mathbf{T}$ and computed the target matrix as $\mathbf{Y} = \mathbf{XT}$. Since we are dealing with robust Procrustes analysis, we added some errors and outliers to $\mathbf{X}$. The procedure of Verboon and Heiser (1992) was followed to add error to the elements in $\mathbf{X}$, proportional to the standard deviations of the columns of $\mathbf{X}$. They also introduced outliers by selecting randomly p rows of $\mathbf{X}$, and multiplying the elements of the selected rows by -10.

Four experimental factors were used in our simulation: (a) the number of rows of $\mathbf{Y}$ and $\mathbf{X}$ ($n = 20, 40$), (b) the number of columns in $\mathbf{Y}$ and $\mathbf{X}$ ($k = 2, 4, 8$), (c) the proportion of error in $\mathbf{X}$ with values $(0, .1, .5, 1)$, and (d) the percentage of outliers in $\mathbf{T}$ ($0\%, 10\%, 20\%$). In this way, 72 different combinations of data sets were obtained. We replicated each of them 100 times, yielding a total of 7200 comparisons. Each comparison of the two algorithms used the same $\mathbf{Y}$, $\mathbf{X}$ and $\mathbf{T}$.

In only 377 out of 7200 runs (5%), the two algorithms stopped at the same local minimum (that was defined by both function values having the same loss at least four equal decimal places). It seems that this form of robust Procrustes analysis has a serious global minimum problem. Comparing the speed for these 377 runs showed that on average our algorithm is about 14 times faster than the one using Kiers' (1997) procedure. Of course, in 95% of the 7200 runs we cannot compare the speed of the two algorithms.

We now turn to the quality of the solutions. Over all 7200 runs, our method reached a lower loss value in all but two runs. As the problem seems to be very prone to local minima, we consider the strategy of 100 random starts and choosing the best solution as a single case. Then, we have 72 different cases here. Our algorithm reached a better minimum in 98% of these cases. In the six cases with zero error and no outliers, a zero loss solution exists. Our algorithm found in two of such cases the zero loss solution, in two other cases a solution

with a slight nonzero loss (.03 and .05), and the two remaining cases were clearly nonzero (.09 and .10). The method based on Kiers' (1997) procedure only found strongly nonzero solutions.

We may conclude that the weighted majorizing algorithm had superior quality solutions. If the two algorithms reach the same local minimum, then our weighted algorithm reaches the solution much faster.

4 Logistic Bi-Additive Models

Logistic models are often used for the analysis of binary data. In this section, we discuss the so-called logistic bi-additive models. For these models, we introduce a new iterative majorization algorithm such that in each iteration a weighted least-squares problem arises, similar to the weighted principal components analysis L_{WPCA} defined in (3). We apply the method from Section 2 and compare the quality and speed of the algorithms.

In our notation, we will follow a similar approach as McCullagh and Nelder (1989) do for generalized linear models. As in principal component analysis, we have data of n individuals on k items in the data matrix $\mathbf{Y}$. The data y_{ij} are binary and indicate, for example, whether a respondent i has given a correct ($y_{ij} = 1$) or incorrect ($y_{ij} = 0$) response to item j. It is common practice to assume that the probability p_{ij} of a correct response is binomially distributed. The mean μ_{ij} of this probability is often modelled by the logistic function $\mu_{ij} = (1 + e^{-\gamma_{ij}})^{-1}$. The elements γ_{ij} are gathered in the $n \times k$ matrix $\mathbf{\Gamma}$ that will be restricted below to be of a special form containing bi-additive terms. Then, the likelihood function becomes

$$L(\mathbf{\Gamma}) = \prod_{ij} \mu_{ij}^{y_{ij}} (1 - \mu_{ij})^{1 - y_{ij}}. \tag{32}$$

that needs to be maximized over $\mathbf{\Gamma}$. Taking minus the log of (32) gives

$$\begin{aligned} -\log L(\mathbf{\Gamma}) &= -\sum_{ij} \left(y_{ij} \log(\mu_{ij}) + (1 - y_{ij}) \log(1 - \mu_{ij}) \right) \\ &= \sum_{ij} \left(y_{ij} \log(1 + e^{-\gamma_{ij}}) + (1 - y_{ij})(\gamma_{ij} + \log(1 + e^{-\gamma_{ij}}) \right) \end{aligned} \tag{33}$$

To find the estimates, it is more convenient to minimize minus the log likelihood in (33) than to maximize the likelihood of (32).

It may be verified that minimizing (33) over the γ_{ij}'s amounts to choosing $\gamma_{ij} = \infty$ if $y_{ij} = 1$ and $\gamma_{ij} = -\infty$ if $y_{ij} = 0$ which is trivial. Therefore, a proper model needs restrictions on the γ_{ij}'s. Here, they are restricted by bi-additive models that may have the following effects:

- the overall mean c,
- main effects for the rows in the n vector $\mathbf{a}$,

- main effects for the columns in the k vector $\mathbf{b}$, and
- the bi-additive interaction effect between rows and columns estimated by the rank p decomposition $\mathbf{UV}'$ with $\mathbf{U}$ of size $n \times p$ and $\mathbf{V}$ of size $k \times p$.

In bi-additive modelling, the final term is always present but any combination of the other effects above can be used. The model with all these effects is given by

$$\mathbf{\Gamma} = \delta_c c\mathbf{1}\mathbf{1}' + \delta_{\mathbf{a}}\mathbf{a}\mathbf{1}' + \delta_{\mathbf{b}}\mathbf{1}\mathbf{b}' + \delta_{\mathbf{UV}'}\mathbf{UV}', \tag{34}$$

where the rank p of $\mathbf{U}$ and $\mathbf{V}$ has to be specified *a priori*, and $\delta_c, \delta_{\mathbf{a}}$, $\delta_{\mathbf{b}}$, and $\delta_{\mathbf{UV}'}$ are indicators that have a value one when the effect is present in the model, and zero otherwise. Whenever the bi-additive term $\mathbf{UV}'$ is omitted (thus $\delta_{\mathbf{UV}'} = 0$), then any combination of the other terms leads to a logistic regression problem. In this paper, we are only interested in models where $\mathbf{UV}'$ is present in the model.

Another special and important example of a bi-additive logistic model is the two-parameter logistic item response model of Birnbaum (1968) in item response theory (IRT). For a connection between generalized linear modelling and IRT, see Mellenbergh (1994). In a two-parameter logistic IRT model, the ability of an individual i is noted by θ_i, the discrimination of an item j by α_j, and the difficulty of item j by β_j. The probability that individual i gives a correct answer to item j is equal to $(1 + e^{-\gamma_{ij}})^{-1}$ with

$$\gamma_{ij} = \alpha_j\theta_i - \beta_j. \tag{35}$$

The two parameter logistic model (35) is exactly equal to choosing $\delta_c = 0, \delta_{\mathbf{a}} = 0, \delta_{\mathbf{b}} = 1$, and $\delta_{\mathbf{UV}'} = 1$ with $p = 1$, that is, $\mathbf{\Gamma} = \mathbf{1}\mathbf{b}' + \mathbf{UV}'$. The equivalence can be seen by expressing the elements of $\mathbf{\Gamma} = \mathbf{1}\mathbf{b}' + \mathbf{UV}'$ as

$$\gamma_{ij} = b_j + u_i v_j \tag{36}$$

which shows that by choosing $\alpha_j = v_j$, $\theta_i = u_i$, and $\beta_j = -b_j$ logistic bi-additive modelling is equivalent to the two parameter logistic IRT model.

Consider a multidimensional version of the two parameter logistic model (see, for example, Béguin & Glas, 2001)

$$\gamma_{ij} = \sum_{s=1}^{p} \alpha_{js}\theta_{is} - \beta_j. \tag{37}$$

In a similar way, (37) can be modelled by bi-additive logistic models by $\mathbf{\Gamma} = \mathbf{1}\mathbf{b}' + \mathbf{UV}'$ with $\mathbf{U}$ and $\mathbf{V}$ rank p, or by

$$\gamma_{ij} = \sum_{s=1}^{p} v_{js}u_{is} + b_j. \tag{38}$$

Note that choosing $\mathbf{\Gamma}$ as in (34) does not give rise to unique estimates. As in analysis of variance, when a main effect $\mathbf{a}$ or $\mathbf{b}$ is estimated alongside with an

Table 3: Overview of effects and degrees of freedom in bi-additive modelling.

	Model $\boldsymbol{\Gamma}$				Degrees of freedom per effect			
	δ_c	$\delta_{\mathbf{a}}$	$\delta_{\mathbf{b}}$	$\delta_{\mathbf{UV}'}$	$c\mathbf{11}'$	$\mathbf{a1}'$	$\mathbf{1b}'$	$\mathbf{UV}'$
a.	1	0	0	0	1			
b.	0	1	0	0		n		
c.	0	0	1	0			k	
d.	0	0	0	1				$np + kp - p^2$
e.	1	1	0	0	1+	$(n-1)$		
f.	1	0	1	0	1+		$(k-1)$	
g.	1	0	0	1	1+			$(np + kp - p^2 - 1)$
h.	0	1	1	0	1+	$(n-1)+$	$(k-1)$	
i.	0	1	0	1		$n+$		$np + (k-1)p - p^2$
j.	0	0	1	1			$k+$	$(n-1)p + kp - p^2$
k.	1	1	1	0	1+	$(n-1)+$	$(k-1)$	
l.	1	1	0	1	1+	$(n-1)+$		$np + (k-1)p - p^2$
m.	1	0	1	1	1+		$(k-1)+$	$(n-1)p + kp - p^2$
n.	0	1	1	1	1+	$(n-1)+$	$(k-1)+$	$(n-1)p + (k-1)p - p^2$
o.	1	1	1	1	1+	$(n-1)+$	$(k-1)+$	$(n-1)p + (k-1)p - p^2$

intercept c, unique main effects are only obtained by imposing additional restrictions on the main effects such as $\mathbf{1}'\mathbf{a} = 0$ and $\mathbf{1}'\mathbf{b} = 0$. Similar nonuniqueness exists if the main effects are estimated alongside with the bi-additive term $\mathbf{UV}'$. If $\mathbf{a}$ is estimated, then we impose the restriction $\mathbf{1}'\mathbf{V} = \mathbf{0}$ and when $\mathbf{b}$ is estimated, the restriction $\mathbf{1}'\mathbf{U} = \mathbf{0}$. Finally, there is always nonuniqueness in the $\mathbf{U}$ and $\mathbf{V}$, as $\mathbf{UV}' = \mathbf{UTT}^{-1}\mathbf{V}'$ for any arbitrary $p \times p$ matrix $\mathbf{T}$. By setting $\mathbf{U}$ equal to n times the left singular vectors, we make $\mathbf{U}$ and $\mathbf{V}$ unique up to a reflection per dimension. Table 3 shows the total degrees of freedom for all models that can be combined from the individual terms of (34). Note that in models (h) and (n) both main effects are estimated, without an explicit mean c. However, implicitly, the main effects do estimate the mean c. Therefore, it is better use (k) instead of (h) and (o) instead of (n). Below, we provide a new majorization algorithm to estimate the terms of the bi-additive logistic model.

4.1 Logistic majorization

In this subsection, we discuss a new iterative majorization algorithm that can be used to minimize minus log likelihood (33) of any logistic function while assuming the binomial distribution. This algorithm yields a majorizing function of the form of weighted PCA (3). Subsequently, we apply the results from Section 2 and derive the updates that use the extra restrictions for the individual terms as indicated in the previous section. Note, that the alternative iterative majorization algorithm for logistic regression proposed by Lange, Hunter, and Yang (2000) cannot be applied directly because it relies on the fact that $\boldsymbol{\Gamma}$ is linear in the parameters to be estimated, which is not the case for bi-additive models.

To derive the new majorization algorithm, consider the terms $\log(1+e^{-\gamma_{ij}})$ for $y_{ij}=1$ and $\gamma_{ij}+\log(1+e^{-\gamma_{ij}})$ for $y_{ij}=0$. Note that both terms are the same up to a reflection of γ_{ij}, that is,

$$\begin{aligned}\gamma_{ij}+\log(1+e^{-\gamma_{ij}}) &= \log(e^{\gamma_{ij}})+\log(1+e^{-\gamma_{ij}})\\ &= \log(e^{\gamma_{ij}})(1+e^{-\gamma_{ij}})\\ &= \log(e^{\gamma_{ij}}+1).\end{aligned}$$

Thus, it is enough to develop a majorizing inequality for the term $\log(1+e^{-\gamma_{ij}})$ only, see Figure 1. For notational convenience, we switch to $f_1(x)=\log(1+e^{-x})$ that needs to be majorized by a quadratic function $g_1(x,y)=a_1x^2-2b_1x+c_1$, with y supporting point, that is, the current estimate. Iterative majorization requires touching at the supporting point, which implies $f_1(y)=g_1(y,y)$ and equal first derivatives $f_1'(y)=g_1'(y,y)$. These two restrictions are not enough to estimate the three parameters a_1, b_1, and c_1 of $g_1(x,y)$. Therefore, we shall impose an additional restriction, that turns out to be convenient, that is, we impose in addition an equal first derivative at $-y$, so that $f_1'(-y)=g_1'(-y,y)$. Note that $f_1'(y)=-(1+e^y)^{-1}$. From these requirements, we derive

$$\begin{aligned}f_1(y) &= g_1(y,y)=a_1y^2-2b_1y+c_1\\ f_1'(y) &= g_1'(y,y)=2a_1y-2b_1\\ f_1'(-y) &= g_1'(-y,y)=-2a_1y-2b_1.\end{aligned}$$

Solving for a_1, b_1, and c_1 gives

$$a_1 = \frac{f_1'(y)-f_1'(-y)}{4y} \tag{39}$$

$$b_1 = a_1y-\tfrac{1}{2}f_1'(y) \tag{40}$$

$$c_1 = f_1(y)-a_1y^2+2b_1y. \tag{41}$$

If $y=0$, then a_1 is not defined. To see what happens to a_1 when y approaches zero, we investigate $\lim_{y\to 0}(f_1'(y)-f_1'(-y))/(4y)$. To do so, we use l'Hôpital's rule, which states that if a limit of a ratio of functions becomes $0/0$, then the ratio of the derivatives of these functions can be used instead. We first rewrite $(f_1'(y)-f_1'(-y))/(4y)$ as

$$\begin{aligned}\frac{f_1'(y)-f_1'(-y)}{4y} &= \frac{-(1+e^y)^{-1}+(1+e^{-y})^{-1}}{4y}\\ &= \frac{-(1+e^y)^{-1}+e^y(1+e^y)^{-1}}{4y}\\ &= \frac{(e^y-1)(1+e^y)^{-1}}{4y}.\end{aligned} \tag{42}$$

The derivative of $(e^y-1)/(1+e^y)$ equals $2e^y/(1+e^y)^2$ and that of $4y$ equals 4. Applying l'Hôpital's rule gives

$$\lim_{y\to 0}\frac{f_1'(y)-f_1'(-y)}{4y}=\lim_{y\to 0}\frac{(2e^y)(1+e^y)^{-2}}{4}=1/8.$$

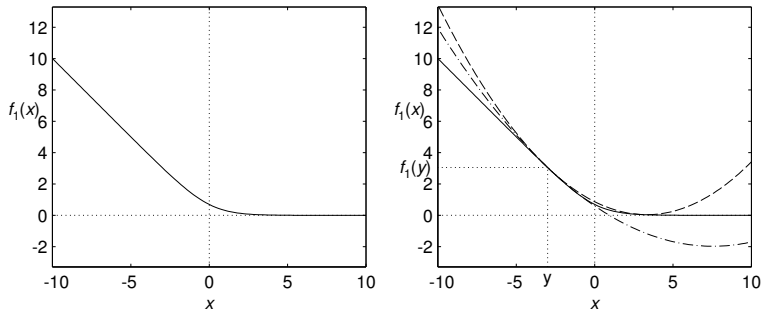

Figure 1: Left panel contains a plot of the function $f_1(x) = \log(1 + e^{-x})$. The right panel shows a $f(x)$ together with the majorizing function $g_1(x, y)$ (dashed line) and a quadratic function used by iterated weighted least squares (IWLS, line with dashes and points).

Thus, for $y = 0$, we define $a_1 = 1/8$. For a_1 defined by (39), b_1 can be simplified into

$$\begin{aligned} b_1 &= a_1 y - \tfrac{1}{2} f_1'(y) = \frac{f_1'(y) - f_1'(-y)}{4y} y - \tfrac{1}{2} f_1'(y) = \frac{f_1'(y) - f_1'(-y)}{4} - \tfrac{1}{2} f_1'(y) \\ &= -\frac{f_1'(y) + f_1'(-y)}{4} = \frac{(1 + e^y)^{-1} + (1 + e^{-y})^{-1}}{4} \\ &= \frac{(1 + e^y)^{-1} + e^y (1 + e^y)^{-1}}{4} = \frac{1}{4}. \end{aligned} \tag{43}$$

Also, at $-y$ we have $f_1(-y) = g_1(-y, y)$ because

$$\begin{aligned} g_1(-y, y) &= a_1 y^2 + 2b_1 y + c_1 = a_1 y^2 + 2b_1 y + f_1(y) - a_1 y^2 + 2b_1 y \\ &= 4b_1 y + f_1(y) = y + f_1(y) = \log e^y + \log(1 + e^{-y}) \\ &= \log e^y (1 + e^{-y}) = \log(e^y + 1) = f_1(-y). \end{aligned}$$

Thus, the function $g_1(x, y)$ touches $f_1(x)$ at y and $-y$ implying that the function values and derivatives are equal at these points.

What remains to be done is to prove that $g_1(x, y) \geq f_1(x)$, or, equivalently, $g_1(x, y) - f_1(x) \geq 0$. Figure 2 shows the function $g_1(x, y) - f_1(x)$. It can be seen that $g_1(x, y) - f_1(x)$ seems to be larger than or equal to zero and that it is exactly zero at $x = y$ and $x = -y$. A rigorous proof is given in the appendix. We call this new approach for minimizing the logistic function *logistic majorization*.

From the results above, we can derive the coefficients of the majorizing function $g_2(x, y) = a_2 x^2 - 2b_2 x + c_2$ for the term $f_2(x) = x + \log(1 + e^{-x}) = \log(e^x + 1)$. We obtain

$$\begin{aligned} a_2 &= \frac{f_1'(y) - f_1'(-y)}{4y} \\ b_2 &= a_2 y - \tfrac{1}{2} f_2'(y) \\ c_2 &= f_2(y) - a_2 y^2 + 2b_2 y. \end{aligned}$$

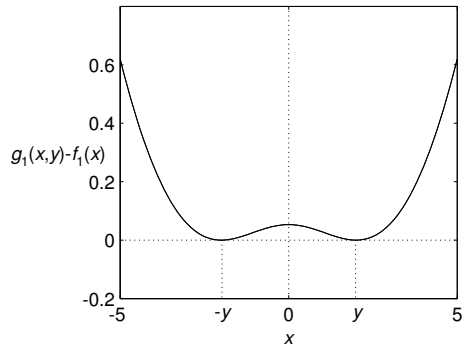

Figure 2: Plot of the function $g_1(x, y) - f_1(x)$. It can be seen that $g_1(x, y) - f_1(x) \geq 0$ and that $g_1(x, y) - f_1(x) = 0$ at $x = y$ and $x = -y$.

Note that $a_1 = a_2$.

A majorizing function for each combination of ij can be obtained as follows. First, we substitute γ_{ij} for x to obtain that $f_1(\gamma_{ij}) \leq g_1(\gamma_{ij}, \gamma_{ij}^0)$ and $f_2(\gamma_{ij}) \leq g_2(\gamma_{ij}, \gamma_{ij}^0)$ where γ_{ij}^0 is the current known estimate of γ_{ij}. Let the subscript ij as in a_{1ij} denote (39) evaluated at $y = \gamma_{ij}^0$. Then (33) is majorized by

$$\begin{aligned} -\log L(\mathbf{\Gamma}) \quad &\leq \quad \sum_{ij} y_{ij}[a_{1ij}\gamma_{ij}^2 - 2b_{1ij}\gamma_{ij} + c_{1ij}] + \\ &\qquad \sum_{ij}(1 - y_{ij})[a_{2ij}\gamma_{ij}^2 - 2b_{2ij}\gamma_{ij} + c_{2ij}]. \end{aligned} \tag{44}$$

This result can be re-expressed as

$$-\log L(\mathbf{\Gamma}) \quad \leq \quad \sum_{ij} w_{ij}(h_{ij} - \gamma_{ij})^2 + c = \mu_{\text{biadd}}(\mathbf{\Gamma}, \mathbf{\Gamma}^0), \tag{45}$$

where

$$\begin{aligned} w_{ij} &= a_{1ij} \\ h_{ij} &= y_{ij}b_{1ij}/a_{1ij} + (1 - y_{ij})b_{2ij}/a_{1ij} \\ c &= \sum_{ij} \left[y_{ij}c_{1ij} + (1 - y_{ij})c_{2ij} - w_{ij}h_{ij}^2\right]. \end{aligned}$$

The majorizing function $\mu_{\text{biadd}}(\mathbf{\Gamma}, \mathbf{\Gamma}^0)$ in (45) is a weighted least-squares problem similar to the one for weighted principal components analysis in (3). Here, too, we can either apply the method of Kiers (1997) directly, or apply the weighted majorization of Section 2. We continue with the latter option. Then, $\mu_{\text{biadd}}(\mathbf{\Gamma}, \mathbf{\Gamma}^0)$ can be majorized again by the quadratic function

$$\text{tr } (\mathbf{R} - \mathbf{\Gamma})'\mathbf{D_m}(\mathbf{R} - \mathbf{\Gamma}) + c, \tag{46}$$

where c is a constant not depending on $\mathbf{\Gamma}$ and

$$r_{ij} = \left[1 - \frac{w_{ij}}{m_i}\right] \gamma_{ij}^0 + \frac{w_{ij}}{m_i} h_{ij}. \tag{47}$$

Finally, we need to specify how an update is obtained for each effect in the bi-additive model. Let us express the quadratic part in (46) as

$$\begin{aligned} \text{tr } (\mathbf{R} - \mathbf{\Gamma})' \mathbf{D_m} (\mathbf{R} - \mathbf{\Gamma}) &= \text{tr } (\mathbf{D_m}^{1/2}\mathbf{R} - \mathbf{D_m}^{1/2}\mathbf{\Gamma})' (\mathbf{D_m}^{1/2}\mathbf{R} - \mathbf{D_m}^{1/2}\mathbf{\Gamma}). \\ &= \text{tr } (\mathbf{D_m}^{1/2}\mathbf{R} - \mathbf{\Gamma}_t)' (\mathbf{D_m}^{1/2}\mathbf{R} - \mathbf{\Gamma}_t). \end{aligned} \tag{48}$$

In (48), we switch a least-squares problem in the diagonal metric $\mathbf{D_m}$ in the parameters $\mathbf{\Gamma}$ to an unweighted least-squares problem with parameters $\mathbf{\Gamma}_t$. Without loss of generality, we impose the bi-additive model directly on $\mathbf{\Gamma}_t$ instead of $\mathbf{\Gamma}$ to get the update for the individual terms. The loss is computed on $\mathbf{\Gamma} = \mathbf{D_m}^{-1/2}\mathbf{\Gamma}_t$. After convergence, the effects are decomposed from $\mathbf{\Gamma}$. During the iterations, the updates are computed as follows. Let $\mathbf{Z} = \mathbf{D_m}^{1/2}\mathbf{R}$.

- If $\delta_c = 1$ then $c = \mathbf{1}'\mathbf{Z}\mathbf{1}/(nk)$ and $\mathbf{Z} = \mathbf{Z} - c\mathbf{1}\mathbf{1}'$.
- If $\delta_{\mathbf{a}} = 1$ then $\mathbf{a} = \mathbf{Z}\mathbf{1}/k$ and $\mathbf{Z} = \mathbf{Z} - \mathbf{a}\mathbf{1}'$.
- If $\delta_{\mathbf{b}} = 1$ then $\mathbf{b} = \mathbf{Z}'\mathbf{1}/n$ and $\mathbf{Z} = \mathbf{Z} - \mathbf{1}\mathbf{b}'$.
- If $\delta_{\mathbf{UV}'} = 1$ then compute the singular value decomposition of $\mathbf{Z}$ by $\mathbf{Z} = \mathbf{K}\mathbf{\Lambda}\mathbf{L}'$. Again, $\mathbf{K}_p$, $\mathbf{\Lambda}_p$, and $\mathbf{L}_p$ denote the first p dimensions of the singular value decomposition. Then, we define $\mathbf{U} = n^{1/2}\mathbf{K}_p$ and $\mathbf{V} = n^{-1/2}\mathbf{L}_p\mathbf{\Lambda}_p$ so that the columns of $\mathbf{U}$ have variance one and are orthogonal.

These updates ensure that uniqueness restrictions outlined above are automatically imposed. After convergence, we compute $\mathbf{\Gamma} = \mathbf{D_m}^{-1/2}\mathbf{\Gamma}_t$ and reconstruct the proper effects by the update procedure above using $\mathbf{Z} = \mathbf{\Gamma}$.

4.2 A small simulation study

Here, we give the results of a small simulation study. Again, we compare the approach by Kiers (1997) in updating $\mu_{\text{biadd}}(\mathbf{\Gamma}, \mathbf{\Gamma}^0)$ with our weighted approach. Remember that $\mu_{\text{biadd}}(\mathbf{\Gamma}, \mathbf{\Gamma}^0)$ was obtained by logistic majorization. A different and standard way to minimize (45) is to apply iterated weighted least squares (IWLS), that also yields a weighted least squares function similar to (45), but with a different definition of w_{ij} (see, for example, McCullagh & Nelder, 1989). IWLS chooses $a_{1ij} = e^{-\gamma_{ij}}/(1 + e^{-\gamma_{ij}})^2$. For this choice of a_{1ij}, b_{1ij} and c_{1ij} can be computed by (40) and (41) respectively. The IWLS function is plotted in Figure 1 by the line with dashes and dots. As the IWLS function crosses $f(x)$, the IWLS algorithm is not guaranteed to converge, although in practice convergence may well be obtained.

Now we have four methods to compare:

1. logistic majorization with the approach by Kiers (1997),

2. logistic majorization with our weighted majorization approach,

3. IWLS with the approach by Kiers (1997), and

4. IWLS majorization with our weighted majorization approach.

In this simulation study, we consider the model with $\delta_c = 0, \delta_{\mathbf{a}} = 0, \delta_{\mathbf{b}} = 1, \delta_{\mathbf{UV}'} = 1$, and $p = 1$ or $p = 2$ so that $\mathbf{\Gamma} = \mathbf{1b}' + \mathbf{UV}'$. This model is equivalent to the two parameter logistic IRT model for $p = 1$ and is a two-dimensional extension for $p = 2$. The data were generated according to this model, that is, we randomly draw $\mathbf{U}, \mathbf{V}$, and $\mathbf{b}$ from the standard normal distribution.

In this study, we varied the factors n (500, 1000), k (50, 100), and p (1, 2). For each combination of n, k, and p, we did five replications. In this way, 40 different data sets were obtained. Because the value of the log likelihood generally will increase if the number of data values nk increases (because the number of terms in the log likelihood increases), we average $-\log L(\mathbf{\Gamma})$ over nk. In this way, for different combinations of n and k, we may expect similar values of the average log likelihood, so that their comparison becomes easier. The iterations were stopped whenever the difference in subsequent average log likelihoods was less than 10^{-8} or the number of iterations exceeded 2000.

Preliminary experimentation with the algorithms showed that if k is small or the number of parameters increases (for example, from $p = 1$ to $p = 2$) then the algorithms may converge slowly. The reason for this to happen is that the parameter space has such a shape that for certain data elements μ_{ij} can get increasingly closer to one (or zero) by letting γ_{ij} tend to ∞ (or $-\infty$). The likelihood function asymptotically approaches zero if γ_{ij} tend to ∞ (or $-\infty$). Thus, if k is too small or the dimensionality too high, then some of the parameters may wander off to large values and the algorithm may require thousands of iterations to obtain reasonable convergence of the parameters, if it is reached at all. However, the changes in the parameter estimates are small during most of the iterations. The early stages of the algorithms show most of the difference. Therefore, we focus on this part and have set a limit to the number of iterations of 2000.

For twelve of the forty data sets (30%) in our experiment, the maximum number of iterations was reached for all of the four methods. Indeed, this situation occurred more often in two dimensions than in one and when n is smaller. For the other cases, there was no significant difference found in the quality of the solution (the average log likelihood) for any of the four methods. However, there was a difference in quality of the solution for the twelve data sets that needed 2000 iterations. It turned out that after 2000 iterations there is hardly any difference between the average log likelihood between the method by Kiers (1997) and weighted majorization. However, a difference was found between logistic majorization and IWLS in favor of the former. A plot of these differences is presented in Figure 3, where the horizontal axis gives the data set and the vertical axis represents the minus log likelihood in deviance of its average for the data set over the four methods. The log likelihood for logistic majorization is presented by a '*' and is connected by a vertical line to the

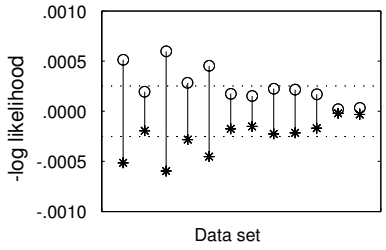

Figure 3: Difference in log likelihood between logistic majorization (*) and IWLS (o) for 12 data sets in deviation of the mean log likelihood for the data set. Because the method based on Kiers' (1997) procedure and by weighted majorization yielded very similar results, their average is taken so that every 'o' and '*' represents the average of the runs for these two methods. The difference is corrected for the average log likelihood of the data set. The dotted lines indicate the averages for IWLS (upper line) and logistic majorization (lower line).

value for IWLS presented by an 'o'. In all twelve cases and after 2000 iterations, logistic majorization finds lower values of minus the log likelihood. On average, the average log likelihood was .00052 better for logistic majorization compared to IWLS.

Thus, logistic majorization yields systematically lower values of $-\log L(\mathbf{\Gamma})$ than IWLS. To see how the algorithms differ during the iterations, we show for a single data set ($n = 500, k = 20$) in Figure 4 for logistic majorization (solid line) and IWLS (dashed line) the difference in $-\log L(\mathbf{\Gamma})$ between the method by Kiers (1997) and weighted majorization as a function of the first thousand iterations. Note that the four different combinations were all started from the same initial values. We can see that the difference is positive implying that the method by Kiers (1997) is somewhat slower than weighted majorization. Also, this difference falls between the sixth and third decimal of the average log likelihood. After iteration 400, the lines are ascending indicating that the difference in $-\log L(\mathbf{\Gamma})$ increases with the iterations.

For the 70% of the data sets that converged, there was again no difference in the number of iterations of the method by Kiers (1997) and weighted majorization. However, there was a significant difference in the number of iterations between logistic majorization and IWLS: the former is on average 1.8 times faster than the latter with a minimum of 1.4 and maximum of 2.0. The mean number of iterations for logistic majorization was 133 (with standard deviation 103) and for IWLS 232 (standard deviation 150). This comparison shows that whenever logistic bi-additive models converge, logistic majorization reaches convergence about 1.8 times faster than IWLS.

From these experiments several conclusions can be drawn. First, the log likelihood function can be flat in cases with too many parameters, yielding a high number of iterations. In those cases, logistic majorization yields better

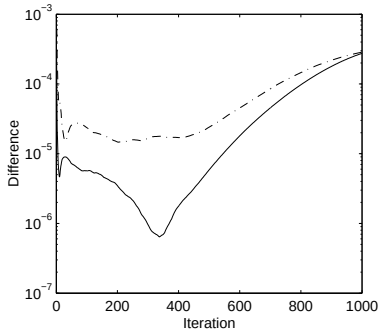

Figure 4: Graph of the difference in $-[\log L(\mathbf{\Gamma})]/(nk)$ between logistic majorization based on Kiers' (1997) procedure and weighted majorization (solid line) and difference between IWLS using Kiers' (1997) procedure and weighted majorization (dashed line).

likelihood values than IWLS. Second, if the algorithms converge, then logistic majorization is about 1.8 times faster than IWLS. There is hardly any difference in quality or speed between weighted majorization and the method by Kiers (1997).

5 Discussion and Conclusions

In this paper, we have proposed a weighted iterative majorization algorithm that improves upon the majorization algorithm by Kiers (1997). Our approach is suited when we are dealing with weighted least-squares decomposition models. We have applied our method to weighted principal components analysis, robust Procrustes analysis, and logistic bi-additive models. Several simulation studies indicate that weighted majorization improves upon the method by Kiers in that it converges faster to a solution (factor 1 to 4 faster) and obtains better quality solutions. In addition, we have proposed a new iterative majorization algorithm for logistic bi-additive models, that yields better quality solutions faster in comparison to iterated weighted least-squares.

Our weighted majorization algorithm only improves for those problems where a diagonally weighted least-squares solution can be easily obtained.

Appendix A: A proof for $g_1(x,y) - f_1(x) \geq 0$

In this appendix, we provide a proof that $g_1(x,y) - f_1(x) \geq 0$. Define the function $h(x,y)$ as

$$h(x,y) = g_1(x,y) - f_1(x) = a_1x^2 - 2b_1x + c_1 - \log(1+e^{-x}). \tag{49}$$

In the sequel, we use results from Section 4.1. In addition, we need

$$f_1(x) = \log(1 + e^{-x}) = \log \frac{1 + e^x}{e^x} = \log(1 + e^x) - x \tag{50}$$

Figure 2 suggests that $h(x, y)$ is symmetric around the point $x = 0$. Here, we provide a mathematical proof for the symmetry of $h(x, y)$, that is,

$$\begin{aligned} h(-x, y) &= g_1(-x, y) - f_1(-x) = a_1x^2 + 2b_1x + c_1 - \log(1 + e^x) \\ &= a_1x^2 + \frac{1}{2}x + c_1 - \log(1 + e^x) = a_1x^2 + \frac{1}{2}x + c_1 - f_1(x) - x \\ &= a_1x^2 - \frac{1}{2}x + c_1 - f_1(x) = a_1x^2 - 2b_1x + c_1 - f_1(x) = h(x, y), \end{aligned} \tag{51}$$

which uses (43) and (50).

The outline of the proof that $h(x, y) \geq 0$ is as follows. First, we investigate what happens to $h(x, y)$ if x tends to ∞ or $-\infty$. Below it is shown that in such cases $h(x, y)$ tends to ∞. Then, we check where $h(x, y)$ has its derivative equal to zero. Two such points are already known, that is, $x = y$ and $x = -y$. Finally, the lowest function values will be attained at a point where $h'(x, y) = 0$. We will prove that this is also at $x = y$ and $x = -y$.

We continue to prove that $\lim_{x\to\infty} h(x, y) = +\infty$ and for this we need to prove that $a_1 > 0$. Equation (42) showed that a_1 can be alternative be represented as

$$a_1 = \frac{(e^y - 1)(1 + e^y)^{-1}}{4y}. \tag{52}$$

For $y > 0$, we have $e^y - 1 > 0$, $1 + e^y > 0$, and $4y > 0$, so that a_1 is positive. For $y < 0$, we have $e^y - 1 < 0$, $1 + e^y > 0$, and $4y < 0$, so that again a_1 is positive. Since at $y = 0$, we defined $a_1 = 1/8$, we have positivity of a_1 for all y. Now, consider

$$\begin{aligned} \lim_{x\to\infty} h(x, y) &= \lim_{x\to\infty} (a_1x^2 - 2b_1x + c_1) - \lim_{x\to\infty} \log(1 + e^{-x}) \\ &= \lim_{x\to\infty} a_1x^2 - \lim_{x\to\infty} \log(1) \\ &= \lim_{x\to\infty} a_1x^2 = \infty, \end{aligned} \tag{53}$$

which uses the fact that $a_1 > 0$. Because of the symmetry of $h(x, y)$ around $x = 0$, we also have that $\lim_{x\to-\infty} h(x, y) = \infty$.

Next, we turn to the derivative of $h(x, y)$ to see where the minima and maxima are attained. Note that the first derivative of $h(x, y)$ equals

$$h'(x, y) = g_1'(x, y) - f_1'(x) = 2a_1x - 2b_1 - \frac{-1}{1 + e^x}. \tag{54}$$

In Section 4.1, it was proved that for $x = -y$ and $x = y$ we have that $f_1(x) = g_1(x, y)$ and $f_1'(x) = g_1'(x, y)$, so that $h'(y, y) = h'(-y, y) = 0$ and $h(y, y) = h(-y, y) = 0$.

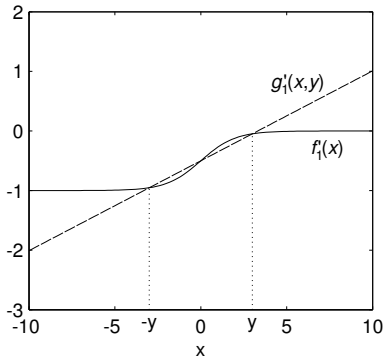

Figure 5: Graph of $g_1'(x, y)$ (solid line) and $f_1'(x)$ (dashed line), which are the two components of the first derivative of $h(x, y)$. At $x = y$, $x = -y$, and $x = 0$ the two derivatives are equal.

We now investigate if there are more than the two stationary points. Analyzing $h'(x, y)$ shows that it is the difference of the straight line $2a_1x - 2b_1$ and $f_1'(x) = -(1 + e^x)^{-1}$. Note that $f_1'(x)$ can also be written as $(1 + e^{-x})^{-1} - 1$ that is equal to the logistic function minus one. Figure 5 shows a graph of the two parts of $h'(x, y)$. We know that the logistic function is concave for $x > 0$ and convex for $x < 0$. As $f_1'(x) = g_1'(x, y)$ at $x = y$ and $x = -y$ and $g_1'(x, y)$ defines a straight line, $g_1'(x, y)$ intersects the shifted logistic curve $f_1'(x)$ at least at those two points. Because of the concavity of the logistic curve for $x > 0$, we can have at most two intersections of the straight line with the curve. Because of the symmetry of $h(x, y)$ around zero, the intersection other than $x = y$ must occur at $x = 0$. This proves that there are at most three stationary points, two at $x = y$ and $x = -y$, the third one at $x = 0$.

We can study the positivity of $h(x, y)$ through $h'(x, y) = g_1'(x, y) - f_1'(x)$ as follows.

- For x in the interval $(-\infty, -y)$, $h'(x, y) < 0$ because the shifted logistic function $f_1'(x)$ is located above the line $g_1'(x, y)$, so that the difference $g_1'(x, y) - f_1'(x) < 0$. At $x = -y$, $h'(x, y) = 0$ because of the majorization requirements in Section 4.1.

- If x lies in the interval $(-y, 0)$, then $h'(x, y) > 0$ because the linear function $g_1'(x, y)$ is located above the logistic function $f_1'(x)$, so that their difference $g_1'(x, y) - f_1'(x) > 0$. At $x = 0$, we have $h'(x, y) = 0$, the second touching point.

- Reverse results occur in the right side of the graph, where $h'(x, y) < 0$ in the interval $(0, y)$, $h'(x, y) > 0$ for x in the interval $(y, +\infty)$, and $h'(x, y) = 0$ at the third stationary point $x = y$.

These properties of the derivatives imply that at $x = y$ and $x = -y$, $h(x, y)$ has a local minimum and at $x = 0$ a local maximum. Because $h(x, y)$ tends

to infinity for $y = -\infty$ and $y = \infty$ and the results on the first derivatives of $h(x, y)$, the global minimum is attained at one of the local minima at $x = y$ or $x = -y$. As $h(y, y) = h(-y, y) = 0$, the global minimum of $h(x, y)$ equals zero, so that $h(x, y) \geq 0$ for all x, which completes the proof.

References

Beaton, A. E., & Tukey, J. W. (1974). The fitting of power series, meaningful polynomials, illustrated on band-spectroscopic data. *Technometrics*, *16*, 147–185.

Béguin, A. A., & Glas, C. A. W. (2001). MCMC estimation and some model-fit analysis of multidimensional IRT models. *Psychometrika*, *66*, 541–562.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), *Statistical theories of mental test scores.* Reading, MA: Addison-Wesley.

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

De Leeuw, J. (1993). *Fitting distances by least squares* (Tech. Rep. No. 130). Los Angeles, California: Interdivisonal Program in Statistics, UCLA.

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

Gabriel, K. R., & Zamir, S. (1979). Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, *21*, 489–498.

Heiser, W. J. (1987). Correspondence analysis with least absolute residuals. *Computational Statistics and Data Analysis*, *5*, 337–356.

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

Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, *35*, 73–101.

Kiers, H. A. L. (1997). Weighted least squares fitting using iterative ordinary least squares algorithms. *Psychometrika*, *62*, 251–266.

Kiers, H. A. L. (2002, in press). Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Computational Statistics and Data Analysis*, *?*, ??–??

Lange, K., Hunter, D. R., & Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, *9*, 1–20.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models.* London: Chapman and Hall.

Mellenbergh, G. J. (1994). Generalized linear item response theory. *Psychological Bulletin*, *115*, 300–307.

Mosteller, F., & Tukey, J. W. (1977). *Data analysis and regression.* Massachusetts: Addison-Wesley.

Ten Berge, J. M. F. (1993). *Least squares optimization in multivariate analysis.* Leiden: DSWO Press, Leiden University.

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

Verboon, P., & Heiser, W. (1992). Resistant orthogonal Procrustes analysis. *Journal of Classification*, *9*, 237–256.