THE MULTIPALS ALGORITHM

Peter Verboon
Ivo A. van der Lans
Willem J. Heiser

Department of Data Theory
University of Leiden

RR-91-04

THE MULTIPALS ALGORITHM

Peter Verboon
Ivo A. van der Lans
Willem J. Heiser

Abstract

A general algorithm is discussed, which can be used to fit four different models, these models are: Principal Component Analysis, Multivariate Multiple Regression, Redundancy Analysis and Canonical Discriminant Analysis. An important extension is the inclusion of loss weights, which makes the optimization procedure non-standard. The algorithm is implemented in the computer program MULTIPALS. Some attention is paid to normalization and rotation problems.

Key words: weighted least squares, PCA, MMRA, RA, CDA, majorization

1. Introduction

In this paper we will describe the algorithm underlying the MULTIPALS method. MULTIPALS is an acronym for MULTIple Projection analysis by Alternating Least Squares. The terms *multiple* and *projection* refer to the fact that a number of different problems are solved by iteratively solving a number of smaller problems of one basic form: a generalized projection. A first sketch of MULTIPALS was given in Heiser *et al.* (1988).
An important feature of MULTIPALS is the presence of so-called *loss weights*. We use the term "loss weights" to distinguish them from other type of weights such as regression weights. The loss weights are fixed and attribute weights to each of the loss components; they can be useful in a variety of situations. For instance, loss weights can be used to

increase the influence on the solution of some variables compared to others or to increase or decrease the influence of some objects. It is possible to use the loss weights to handle missing data.

The program is very general, for it can be used to analyze data by choosing from four different techniques or models. These models are: Principal Component Analysis (PCA), Multivariate Multiple Regression Analysis (MMRA), Redundancy Analysis (RA) and Canonical Discriminant Analysis (CDA).
From these different models we have taken the PCA model to illustrate the main lines of the algorithm. After the general outline of the algorithm is given, relations with the other models are briefly discussed.
It is also explained why particular choices with respect to normalization restrictions are made. Furthermore, it is shown how some problems arising from other possible normalizations, which are not incorporated in the present version, can technically be solved.

The general MULTIPALS loss function is written as:

$$\sigma(\mathbf{Q},\mathbf{X},\mathbf{A},\mathbf{C}) = \sum_{j=1}^{m} (\mathbf{q}_j - \mathbf{X}\mathbf{A}\mathbf{c}_j)'\mathbf{V}_j(\mathbf{q}_j - \mathbf{X}\mathbf{A}\mathbf{c}_j), \qquad (1)$$

where $\mathbf{Q} = \{q_{ij}\}$ is an nxm matrix of scores for n objects on m optimally transformed criterion variables; the criterion variables are constrained as $\mathbf{q}_j'\mathbf{q}_j = n$. $\mathbf{X} = \{x_{ik}\}$ is an nxp matrix of scores for the same n objects on p predictor variables (or principal components); $\mathbf{A} = \{a_{kl}\}$ is a pxr matrix of coefficients; $\mathbf{C} = \{c_{jl}\}$, or $\{c_{jk}\}$ in the case of PCA, is an mxr matrix of coefficients; $\mathbf{V}_j =$ is a diagonal nxn matrix of loss weights corresponding to the observations on the jth criterion variable.

The scores in $\mathbf{q}_j$ are transformations of the original scores, where the transformations depend on the measurement levels chosen for the variables. The measurement restrictions can be seen as projections on cones, which means that we project each unrestriced $\mathbf{q}_j$ on its cone of admissible transformations (Gifi, 1990). After convergence of the algorithm, the transformations are optimal with respect to the loss criterion, formulated in (1). The program is capable of dealing with nominal, ordinal (incorporating both the primary and the secondary approach to ties) and numerical measurement levels.

2. Principal Components Analysis

In the PCA model the matrices **X** and **A** from (1) are taken together as one matrix, denoted as **X** (*nxp*). In other words we assume $r = p$, so $\mathbf{A} = \mathbf{I}_p$. The matrix **C**' contains the usual component loadings when the loss weights matrices are identity matrices, thus when we have $\mathbf{V}_j = \mathbf{I}_n$ for each variable *j*:. So the PCA loss function, which is called *join loss* in Gifi (1990, section 4.2) is written as:

$$\sigma(\mathbf{Q},\mathbf{X},\mathbf{C}) = \sum_{j=1}^{m} (\mathbf{q}_j - \mathbf{X}\mathbf{c}_j)'\mathbf{V}_j(\mathbf{q}_j - \mathbf{X}\mathbf{c}_j). \qquad (2)$$

The presence of the loss weights matrices $\mathbf{V}_j$ the loss function in (2) amounts to an extension of the classical PCA problem. Since we also allow different types of quantifications, including missing data, it is also an extension of the PRINCIPALS method of Young, Takane and De Leeuw (1978).

Gifi (1990) distinguishes join-loss from meet-loss, and shows how meet-loss functions can be used to define all sorts of combinations of variables, including principal components; in this case the method is called PRINCALS. The difference between the two ways of defining the objective of analysis can be described as follows. In join-loss the scores collected in $\mathbf{q}_j$ are approximated by projecting the rows of **X** (the object scores) on a specific direction in $\mathbf{c}_j$ in the *p*-dimensional space. Thus, join-loss involves the approximation of an m-dimensional set of scores with bilinear forms in reduced dimensionality. By contrast, meet-loss is not based on the idea of dimension reduction, but on dimension matching. The low-rank configurations $\mathbf{q}_j\mathbf{c}_j'$ are mutually compared by the introduction of the comparison configuration **X**, yielding residuals of the form $\mathbf{X} - \mathbf{q}_j\mathbf{c}_j'$, the sum of squares of which is averaged across variables. It is not hard to show that in the unweighted case join-loss and meet-loss give identical results for numerical variables. However, for other measurement levels and in the presence of loss weights the two loss functions will generally give different results.

In (2), we assume the diagonal weights matrices to be fixed and known. The algorithm to minimize (2) consists of three basic steps; each updates one set of parameters. These steps are alternatingly repeated until some convergence criterion has been reached. The convergence criterion is defined on the loss, that is, the program stops when the loss from two consecutive steps is smaller than a preset constant.

Throughout this process we keep the columns of $\mathbf{Q}$ normalized, as was indicated by (1), while the other set of parameters have no normalization restrictions during the iterations.

2.1 Updating the quantifications of the criterion variables

To avoid degenerate solutions we have to normalize the columns of $\mathbf{Q}$. However, by the presence of the loss weights it is not immediately clear which normalization should be used. There are three possibilities:

(i)	$\mathbf{q}_j'\mathbf{q}_j = n$	with	$\mathbf{q}_j'\mathbf{u} = 0.$
(ii)	$\mathbf{q}_j'\mathbf{V}_j\mathbf{q}_j = n$	with	$\mathbf{q}_j'\mathbf{V}_j\mathbf{u} = 0.$
(iii)	$\mathbf{q}_j'\mathbf{V}_j\mathbf{q}_j = \mathbf{u}'\mathbf{V}_j\mathbf{u}$	with	$\mathbf{q}_j'\mathbf{V}_j\mathbf{u} = 0.$

The vector $\mathbf{u}$ denotes the unit vector and keeping the variables orthogonal to it is equivalent to keeping them centered. Before considering the differences between the three possible normalizations we will first describe the general procedure for updating $\mathbf{Q}$.

We start computing the unrestricted updates for $\mathbf{q}_j$ denoted as $\mathbf{q}_j^0$, given by

$$\mathbf{q}_j^0 = \mathbf{X}\mathbf{c}_j. \tag{3}$$

Next we must find $\mathbf{q}_j$ that satisfies the normalization and transformation restrictions. In addition to the measurement restriction we also require for each variable one of the above mentioned normalizations.To find this restricted $\mathbf{q}_j$ we have to solve for each variable the general problem:

$$\min\ (\mathbf{q}_j - \mathbf{q}_j^0)'\mathbf{V}_j(\mathbf{q}_j - \mathbf{q}_j^0), \tag{4}$$

subject to the measurement and normalization contraints. Because some points may be weighted more strongly than others, the optimal solution to (4) is not simply found by a rescaling of the unrestricted variable, if we require the normalization $\mathbf{q}_j'\mathbf{q}_j = n$. So, for normalization (i) the presence of the matrix $\mathbf{V}_j$ makes this problem non-standard, because the normalization and the loss are defined in different metrics. However, in Heiser (1987) a general majorization procedure is proposed to solve problems of the kind presented in (4) for any positive semi-definite matrix $\mathbf{V}_j$. The present problem is even somewhat simpler,

since $\mathbf{V}_j$ is diagonal. The procedure in Heiser (1987) is iterative. Using this iterative approach, we derive for our problem adjusted updates $\mathbf{q}_j^+$ as:

$$\mathbf{q}_j^+ = \tilde{\mathbf{q}}_j + 1/\beta\, \mathbf{V}_j(\mathbf{q}_j^0 - \tilde{\mathbf{q}}_j) \tag{5}$$

where in general β is chosen as the largest eigenvalue of $\mathbf{V}_j$ (which is for a diagonal $\mathbf{V}_j$, of course, the largest element of $\mathbf{V}_j$) and $\tilde{\mathbf{q}}_j$ is the previous estimate of $\mathbf{q}_j$ satisfying the restrictions. Next, these adjusted updates are projected upon the cone of admissible transformations and normalized. The process of updating, cone projection and normalization is repeated until convergence of the sums of squares for the $\mathbf{q}_j$. This is called the first inner iteration loop.

So, since the general optimization problem is defined in the weighted metric, it follows that if we require normalization (i), which is defined in the unweighted metric, the complete procedure mentioned above, will be necessary. This is a disadvantage compared to the normalizations (ii) and (iii), for which we don't need inner iterations, because here the normalization is in the same metric as the optimization problem. So, with these type of normalizations the normalization is just a rescaling of the variable. There is also another undesirable feature to this first normalization. If we have identical rows in the data matrix, these cannot be replaced by a single row with weight 2, because this turns out not to be equivalent to each other.

The second normalization is not very useful, because the loss weights cannot be used to weight some variables more heavily than others. In Appendix I it is shown that weighting a variable by a constant will change $\mathbf{q}_j$ and $\mathbf{c}_j$ only by a constant while the optimal $\mathbf{X}$ remains unchanged. This invariance is basically due to the fact that the measurement restrictions always have the form of cones; that is, if $\mathbf{q}_j$ satisfies the restrictions, then $k\mathbf{q}_j$ satisfies the restrictions as well, for any positive scalar factor k.

Now consider the third normalization, in which case we normalize the weighted variables to the sum of their weights. Problem (4) can now be written in a different form as:

$$\min\ (\mathbf{q}_j - \mathbf{q}_j^0)'(\mathbf{q}_j - \mathbf{q}_j^0), \tag{6}$$

with $\mathbf{q}_j = \mathbf{V}_j^{1/2}\mathbf{q}_j$ and $\mathbf{q}_j^0 = \mathbf{V}_j^{1/2}\mathbf{q}_j^0$ and the normalization: $\mathbf{q}_j'\mathbf{q}_j = \mathbf{u}'\mathbf{V}_j\mathbf{u}$. The weighted variables are thus considered as new variables, in this way the problem has become

standard, because the normalization is in the same metric as the (weighted) variables, and therefore no majorization is necessary anymore.
After some experimentation, we have chosen for this third normalization in MULTIPALS, because it incorporates the weights in a natural way. Assigning a weight k to all observations on a particular variable or object is equivalent to copying this variable or object k times. Furthermore, the weights can now be used to deal with missing data, by assigning zero weight to each missing observation in the data matrix. And finally there is no inner iterative majorization necessary.

2.2 Updating the Principal Scores

In the MULTIPALS algorithm we have chosen for a dimension-wise strategy of fitting the matrix $\mathbf{X}$. We will use the following special notation to arrive at a convenient form of the loss function:

$$\mathbf{X}_{(-k)} = \mathbf{X} - \mathbf{x}_k\mathbf{e}_k', \tag{7a}$$

$$\mathbb{q}_j = \mathbf{q}_j - \mathbf{X}_{(-k)}\mathbf{c}_j, \tag{7b}$$

$$\mathbf{q}_j - \mathbf{X}\mathbf{c}_j = \mathbb{q}_j - \mathbf{x}_k\mathbf{e}_k'\mathbf{c}_j, \tag{7c}$$

where the vector $\mathbf{e}_k$ is the k^{th} column of the identity matrix. In (7a) we define a matrix $\mathbf{X}_{(-k)}$ that does not depend on the k^{th} component. The vector $\mathbb{q}_j$ defined in (7b) equals that part of $\mathbf{q}_j$ that is based on the k^{th} component of $\mathbf{X}$. Combining (7a) and (7b) yields equality (7c). Furthermore we may simplify the vector multiplication $\mathbf{e}_k'\mathbf{c}_j$ to one single element: c_{jk}. Now we are able to write the general problem in the form:

$$\sigma(\mathbf{x}_k) = \sum_j(\mathbb{q}_j - \mathbf{x}_k c_{jk})'\mathbf{V}_j(\mathbb{q}_j - \mathbf{x}_k c_{jk}). \tag{8}$$

Differentiating (8) with respect to $\mathbf{x}_k$ yields the unrestricted $\mathbf{x}_k^0$ which satisfies the following stationary equation:

$$\sum_j c_{jk}^2\mathbf{V}_j\,\mathbf{x}_k^0 = \sum_j c_{jk}\mathbf{V}_j\mathbb{q}_j. \tag{9}$$

If we want each $\mathbf{x}_k$ in deviation from its mean for any choice of $\mathbf{V}_j$ we need iterative majorization. First consider the problem with one particular variable involved, in other

words we skip the summation over the variables for the moment. This yields after rewriting (8) the following loss function:

$$\sigma_j(\mathbf{x}_k) = (\mathbf{q}_j - c_{jk}\mathbf{x}_k)'\mathbf{V}_j(\mathbf{q}_j - c_{jk}\mathbf{x}_k). \tag{10}$$

Next we can decompose (10) into the following form:

$$\sigma_j(\mathbf{x}_k) = \sigma_j(\mathbf{x}_k^0) + (\mathbf{x}_k^0 - \mathbf{x}_k)'c_{jk}^2\mathbf{V}_j(\mathbf{x}_k^0 - \mathbf{x}_k). \tag{11}$$

The proof for the validity of this decomposition is given in the Appendix II; the present problem is actually somewhat simpler than the more general problem, given in the appendix, since the matrix $\mathbf{C}$ in the appendix is replaced by the scalar c_{jk} in (10). Finally, the summation over the variables yields

$$\sigma(\mathbf{x}_k) = \sigma(\mathbf{x}_k^0) + (\mathbf{x}_k^0 - \mathbf{x}_k)'\mathbf{V}_k(\mathbf{x}_k^0 - \mathbf{x}_k), \tag{12}$$

where $\mathbf{V}_k$ is the aggregated matrix defined as $\Sigma_j c_{jk}^2\mathbf{V}_j$. The second part of this function is identical to (4) in form, so the solution can also be given analogously, by the updating formula

$$\mathbf{x}_k^+ = \tilde{\mathbf{x}}_k + 1/\gamma\, \mathbf{V}_k(\mathbf{x}_k^0 - \tilde{\mathbf{x}}_k). \tag{13}$$

Here γ is chosen as the largest element of $\mathbf{V}_k$ and $\tilde{\mathbf{x}}_k$ is the previous centered estimate. Repeatingly computing updates and centering leads to convergence. This is the second inner iteration loop.

Note that we do not have to normalize $\mathbf{x}_k$ at this stage of the algorithm, since $\mathbf{x}_k$ cannot become arbitrarily small, because $\mathbf{q}_j$ is of fixed length. A possible normalization of $\mathbf{X}$ is, unlike $\mathbf{Q}$, for identification purposes only.

If the $\mathbf{x}_k$ should not be explicitly required in deviation from the mean then we are directly finished when we solve the stationary equation (9). So, the second inner iteration step is not necessary then.

In the special case that the matrices $\mathbf{V}_j$ are the same for all variables, it follows directly that $\mathbf{u}'\mathbf{V}_j\mathbf{x}_k^0 = 0$. From the stationary equation we write

$$\mathbf{x}_k^0 = (\Sigma_j c_{jk}^2\mathbf{V}_j)^{-1}\Sigma_j c_{jk}\mathbf{V}_j\mathbf{q}_j\,, \tag{14}$$

with q_j defined as in (7b). If we assume all columns of $\mathbf{X}$ and all $\mathbf{q}_j$ in deviation from the weighted mean, then the new unrestricted $\mathbf{x}_k^0$ is also in deviation from the weighted mean. In MULTIPALS we have chosen not to impose centering of $\mathbf{x}_k$ as a restriction; however, as we have seen above, in special cases $\mathbf{x}_k$ will be in deviation from the mean anyway.

2.3 Updating the component loadings

From (2) it follows that the vectors $\mathbf{c}_j$ can be computed as weighted regression coefficients, that is

$$\mathbf{c}_j = (\mathbf{X}'\mathbf{V}_j\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}_j\mathbf{q}_j \tag{15}$$

However, the inverse of $\mathbf{X}'\mathbf{V}_j\mathbf{X}$ can sometimes be time-consuming to compute, therefore we follow the same procedure as in section 2.2, thus we compute $\mathbf{C}$ dimension-wise. Again we will use the special notation to arrive at a convenient form of the problem. However, from a separation of loss in terms of columns of $\mathbf{Q}$, as in (2), we switch to a separation of loss in terms of rows of $\mathbf{Q}$. The following notation is used:

$$\mathbf{C}_{(-k)} = \mathbf{C} - \mathbf{e}_k\mathbf{c}_k', \tag{16a}$$

$$q_i = \mathbf{q}_i - \mathbf{C}'_{(-k)}\mathbf{x}_i, \tag{16b}$$

$$\mathbf{q}_i - \mathbf{C}'\mathbf{x}_i = q_i - \mathbf{c}_k\mathbf{e}_k'\mathbf{x}_i = q_i - \mathbf{c}_k x_{ik}. \tag{16c}$$

This brings us to the following form of the loss function:

$$\sigma(\mathbf{c}_k) = \Sigma_i(q_i - \mathbf{c}_k x_{ik})'\mathbf{V}_i(q_i - \mathbf{c}_k x_{ik}). \tag{17}$$

where $\mathbf{V}_i$ is a diagonal (mxm) weights matrix with the weights for each row of the data matrix on its main diagonal. Differentiating (17) with respect to $\mathbf{c}_k$ yields the following stationary equation:

$$\Sigma_i x_{ik}^2\mathbf{V}_i\,\mathbf{c}_k = \Sigma_i x_{ik}\mathbf{V}_i q_i. \tag{18}$$

Since there are no restrictions on $\mathbf{C}$ the problem is a simple one:

$$\mathbf{c}_k' = \Sigma_i x_{ik}\mathbf{V}_i q_i / \Sigma_i x_{ik}^2\mathbf{V}_i, \tag{19}$$

so the computation of $\mathbf{C}$ requires no additional majorization steps.

2.4 Identification of the Principal Components

Having found $\mathbf{C}$, we return to step 1 and cycle through the above steps until the solution stabilizes. When we have reached convergence, we rotate the columns of $\mathbf{X}$ into the principal axes orientation. This explicit rotation of $\mathbf{X}$ is necessary, since there is a rotation and scaling indeterminacy in the products $\mathbf{X}\mathbf{c}_j$ in (2). This identification is done successively by first minimizing for the first component

$$\sigma(\mathbf{x}_1; \mathbf{c}_1) = \sum_j (\mathbf{q}_j - c_{j1}\mathbf{x}_1)'\mathbf{V}_j(\mathbf{q}_j - c_{j1}\mathbf{x}_1). \tag{20}$$

This is a weighted criss-cross regression problem (cf. Gabriel and Zamir, 1979). Note that $\mathbf{Q}$ is kept fixed in this stage. With these newly found $\mathbf{c}_1$ and $\mathbf{x}_1$ we compute the matrix of residuals: $\mathbf{Q}_{(-1)} = \mathbf{Q} - \mathbf{x}_1\mathbf{c}'_1$, and use this deflated matrix to compute the second component by minimizing

$$\sigma(\mathbf{x}_2; \mathbf{c}_2) = \sum_j (\mathbf{q}_{(-1)j} - c_{j2}\mathbf{x}_2)'\mathbf{V}_j(\mathbf{q}_{(-1)j} - c_{j2}\mathbf{x}_2). \tag{21}$$

We continue this process until we have computed all components. The minimization of (20) and (21) is simply applying alternating least squares. However, if we restrict the components to be orthogonal, that is $\mathbf{x}_k'\mathbf{x}_l = 0$, for $(k \neq l)$, then we also need a majorization step comparable with (12) and (13) in section 2.3, because this orthogonality restriction is in a different metric than the original problem.

3. Projection onto the Prediction Subspaces

In the sections to follow, the columns of $\mathbf{X}$ are predictor variables, which implies additional measurement restrictions upon the scores in $\mathbf{X}$. These restrictions require an extra step when updating the columns of $\mathbf{X}$. However, this does not complicate the problem as we have defined it so far, for we can still use the updates defined in (13). After the unrestricted updating, we project the variables on their cones and normalize them. In fact, the PCA problem can now be seen as a special case of the situation where the $\mathbf{x}_k$ are observed and known up to some class of transformations.

3.1 Multivariate Multiple Regression Analyis (MMRA)

For the MMRA problem we may also use loss function (2), however, the matrix **X** is now a matrix with observed predictor variables. Like in the algorithm for PCA, we alternatingly update each set of parameters **Q**, **X** and **C**.
In MMRA, the computation of the columns of **Q** is the same as in PCA. As already mentioned above, we have an additional optimal scaling step for the columns of **X**. This implies that we have to project the intermediate quantities defined in (14) on the cone of admissible transformations. This process yields no additional difficulties compared to the solutions described in section 2.2. Furthermore, there are no restrictions upon the matrix **C**. After convergence we are basically finished, because there is no rotation of **X** afterwards, and consequently also no correction of **C**.

3.2 Redundancy Analysis (RA)

For the RA model we return to loss function (1). We now have $\mathbf{A} \neq \mathbf{I}$, which yields one extra step in the algorithm. So the algorithm now consists of four basic steps, but these steps yield no additional problems. For fixed **A**, (1) becomes of the form (2) with the reduced space **XA** as the predictor set rather than **X** itself. For fixed **Q**, **X** and **C** there are several options to compute **A**. We could choose to compute **A** row-wise (per predictor variable) or column-wise (per dimension). However, both approaches hold the computation of the inverse of a matrix that in general will not be diagonal. Since this might be time-consuming, we prefer a third approach: the element-wise fitting of **A**. To show the convenience of this approach, we will use a similar notation as in (7) and (16).

$$\mathbf{A}_{(-kl)} = \mathbf{A} - \mathbf{e}_k a_{kl} \mathbf{e}_l', \tag{22a}$$

$$\mathbf{q}_{lj} = \mathbf{q}_j - \mathbf{X}\mathbf{A}'_{(-kl)}\mathbf{c}_j, \tag{22b}$$

$$\mathbf{q}_j - \mathbf{X}\mathbf{A}'\mathbf{c}_j = \mathbf{q}_{lj} - \mathbf{X}\mathbf{e}_k a_{kl} \mathbf{e}_l' \mathbf{c}_j = \mathbf{q}_{lj} - \mathbf{x}_k a_{kl} c_{jl}, \tag{22c}$$

where $\mathbf{e}_k$ and $\mathbf{e}_l$ are respectively columns from $\mathbf{I}_p$ and $\mathbf{I}_r$. This brings us to the following form of the loss function:

$$\sigma(a_{kl}) = \Sigma_j(\mathbf{q}_j - \mathbf{x}_k a_{kl} c_{jl})'\mathbf{V}_j(\mathbf{q}_j - \mathbf{x}_k a_{kl} c_{jl}). \quad (23)$$

Differentiating (23) with respect to the scalar a_{kl} yields the following stationary equation:

$$\Sigma_j c_{jl}^2 \mathbf{x}_k'\mathbf{V}_j\mathbf{x}_k a_{kl} = \Sigma_j \mathbf{q}_j'\mathbf{V}_j\mathbf{x}_k c_{jl}. \quad (24)$$

Since there are no restrictions on **A** the problem is a simple one:

$$a_{kl} = \frac{\Sigma_j \mathbf{q}_j'\mathbf{V}_j\mathbf{x}_k}{\Sigma_j c_{jl}^2 \mathbf{x}_k'\mathbf{V}_j\mathbf{x}_k}. \quad (25)$$

We repeatedly cycle through all elements of **A** until convergence. So the computation of **A** requires no additional majorization steps.

After convergence of the algorithm the matrices **A** and **C** are not identified for we can always write **AT'SC'** = **AC'** for any pair of full-rank matrices **T** and **S** with the property **T'S** = **I**. A solution for this rotational indeterminacy will be discussed in section 4.

In MULTIPALS the following names are used for the different matrices: the elements of the matrix **XA** are called object scores, **A** contains the canonical weights, **C** the component loadings and finally the elements of **AC'** are called regression weights.

3.3 Canonical Discriminant Analysis (CDA)

For a better understanding of how a solution for the CDA model is computed by minimizing (1), we will start with the discussion of the unweighted problem

$$\sigma(\mathbf{Q},\mathbf{X},\mathbf{A},\mathbf{C}) = \text{tr}\,(\mathbf{Q} - \mathbf{XAC}')'(\mathbf{Q} - \mathbf{XAC}'). \quad (26)$$

There is a difference between CDA and RA in the nature of the matrix **Q**. In the CDA model, the criterion part of the problem consists of an indicator matrix **G**, which partitions the set of objects in mutually exclusive groups. The matrix **G** is post-multiplied with the diagonal matrix $\mathbf{D}^{-1/2}$, where **D** is defined as **G'G**. So for CDA, (26) becomes

$$\sigma(\mathbf{X},\mathbf{A},\mathbf{C}) = \text{tr}\,(\mathbf{GD}^{-1/2} - \mathbf{XAC}')'(\mathbf{GD}^{-1/2} - \mathbf{XAC}'), \quad (27)$$

and $\mathbf{D}^{-1/2}\mathbf{G'GD}^{-1/2} = \mathbf{I}$ by definition.

The classical problem in CDA is defined in terms of maximizing per dimension a ratio of sums of squares (variances):

$$\psi = \frac{s^2_{between}}{s^2_{within}}, \tag{28}$$

which is actually maximizing the between variance, since the within variances per dimension are usually normalized to 1. Since in CDA an orthogonal *r*-dimensional solution is required, that is we are seeking *r* discriminant functions, we obtain *r* different ψ values, that can be placed on the main diagonal of the diagonal matrix Ψ, in which the diagonal elements are in descending order. (See e.g. Van de Geer, 1986; ch. 16). In the MULTIPALS terminology the sums of squares can be written as follows:

$$\text{BETWEEN: } \mathbf{A'X'GD^{-1}G'XA}, \tag{29}$$

$$\text{WITHIN: } \mathbf{A'X'XA - A'X'GD^{-1}G'XA = W}. \tag{30}$$

It can be verified that after convergence and rotation of the minimization problem stated in (26) the matrix **W** is diagonal (see Appendix III); now, since in CDA this **W** is required to be equal to **I**, which can easily be realized by changing **A** into:

$$\mathbf{A^* = AW^{-1/2}}. \tag{31}$$

The loss may not increase, so we adjust **C** as well by

$$\mathbf{C^* = CW^{1/2}}. \tag{32}$$

By minimizing the MULTIPALS loss criterion, we actually maximize tr Ψ, the discriminant criterion. This can quite easily be seen.

Eliminating **C** from loss function (27), and substituting for **C** its least squares estimate, yields

$$\sigma(\mathbf{X,A,^*}) = \text{tr } \mathbf{I} - \text{tr } (\mathbf{D^{-1/2}G'XA(A'X'XA)^{-1}A'X'GD^{-1/2}}). \tag{33}$$

Minimizing (33) is equivalent to maximizing the second term at the right hand side, which after cyclic permutation of its elements becomes tr $(\mathbf{(A'X'XA)^{-1}A'X'GD^{-1}G'XA})$. By replacing **A** by $\mathbf{A^*}$ and using equality (30) and (31), we obtain:

$$\sigma(\mathbf{X},\mathbf{A},\mathbf{C}) = \text{tr}\,\mathbf{I} - \text{tr}\,((\mathbf{I} + \mathbf{A}^{*\prime}\mathbf{X}'\mathbf{G}\mathbf{D}^{-1}\mathbf{G}'\mathbf{X}\mathbf{A}^{*})^{-1}\mathbf{A}^{*\prime}\mathbf{X}'\mathbf{G}\mathbf{D}^{-1}\mathbf{G}'\mathbf{X}\mathbf{A}^{*}), \quad (34)$$

which is equal to maximizing tr $\mathbf{A}^{*\prime}\mathbf{X}'\mathbf{G}\mathbf{D}^{-1}\mathbf{G}'\mathbf{X}\mathbf{A}^{*}$, which is exactly the discriminant criterion, such as given by (28) and (29), under the required restrictions.

Entering loss weights

Now let's examine the problem with loss weights added. We will assume that each group will be weighted equally, that is $\mathbf{V}_j = \mathbf{V}$ for j =1,...,*m*. The weighted loss function with an indicator matrix becomes

$$\sigma(\mathbf{A},\mathbf{C}) = \text{tr}\,(\mathbf{G}\mathbf{S}^{-1/2} - \mathbf{X}\mathbf{A}\mathbf{C}')'\mathbf{V}(\mathbf{G}\mathbf{S}^{-1/2} - \mathbf{X}\mathbf{A}\mathbf{C}'), \quad (35)$$

with the property $\mathbf{S}^{-1/2}\mathbf{G}'\mathbf{V}\mathbf{G}\mathbf{S}^{-1/2} = \mathbf{I}$ where $\mathbf{S}$ is defined as $\mathbf{G}'\mathbf{V}\mathbf{G}$. The diagonal matrices of weighted within and between sums of squares become:

$$\text{BETWEEN:}\quad \mathbf{A}'\mathbf{X}'\mathbf{V}\mathbf{G}\mathbf{S}^{-1}\mathbf{G}'\mathbf{V}\mathbf{X}\mathbf{A}, \quad (36)$$

$$\text{WITHIN:}\quad \mathbf{A}'\mathbf{X}'\mathbf{V}\mathbf{X}\mathbf{A} - \mathbf{A}'\mathbf{X}'\mathbf{V}\mathbf{G}\mathbf{S}^{-1}\mathbf{G}'\mathbf{V}\mathbf{X}\mathbf{A} = \mathbf{W}. \quad (37)$$

The $\mathbf{A}$ and $\mathbf{C}$ can be transformed to $\mathbf{A}^{*}$ and $\mathbf{C}^{*}$ like in (31) and (32). Furthermore, as in the unweighted case, it is obvious that by working out (35), it can be seen that the minimization of $\sigma(\mathbf{A},\mathbf{C})$ is equivalent to maximizing the weighted BETWEEN sums of squares. So we have shown that the CDA solution can easily be obtained from the RA solution with additional normalization conditions for $\mathbf{A}$ and $\mathbf{C}$ in order to satisfy the restrictions that characterize the CDA problem.

4. Rotations and Normalizations

After convergence of the algorithm, the minimum of (1) is attained. However, there is still the freedom left to give the solution a different orientation in the space. We may use this freedom by applying a rotation to obtain some agreeable properties for the rotated solution. Furthermore, we may normalize the variables or principals components to enhance interpretability, that is, different analyses of the same (type of) data become comparable. For PCA, this identification process has already been mentioned in section 2.4. Note that these rotations and normalizations have no effect upon the overall loss.

We may distinguish three steps in the identification process. First the predictor variables are normalized to n. Thus, for all models, except PCA, we have for $k = 1,...,p$.

$$\mathbf{x}_k'\mathbf{x}_k = n. \tag{38}$$

Next we adjust $\mathbf{C}$ (in MMRA) or $\mathbf{A}$ (in RA and CDA) to keep the total sum of squares of the residuals the same. This is the first step; for the MMRA model we are finished.

The second step is to re-distribute the 'explained sum of squares' over the object scores, in such a way that they are in principal axes orientation. So for PCA the matrix $\mathbf{X}$ and for RA and CDA the matrix $\mathbf{XA}$ is rotated. In section 2.4 it has been explained how this can be done.

The third step is a normalization step. We distinguish between two different situations: first, equal weighting of variables, and second differential weighting. When the variables are equally weighted, the problem becomes

$$\sigma(\mathbf{A},\mathbf{C}) = \mathrm{tr}\,(\mathbf{Q}^* - \mathbf{X}^*\mathbf{A}\mathbf{C}')'(\mathbf{Q}^* - \mathbf{X}^*\mathbf{A}\mathbf{C}'), \tag{39}$$

with $\mathbf{Q}^* = \mathbf{V}^{1/2}\mathbf{Q}$ and $\mathbf{X}^* = \mathbf{V}^{1/2}\mathbf{X}$. The matrix $\mathbf{X}^*\mathbf{A}$ ($\mathbf{X}^*$ in PCA) is in principal axes orientation. There are two classical normalizations:

(a) $\mathbf{A}'\mathbf{X}^{*\prime}\mathbf{X}^*\mathbf{A} = n\mathbf{I}$ (RA),
$\mathbf{X}^{*\prime}\mathbf{X}^* = n\mathbf{I}$ (PCA),

(b) $\mathbf{C}'\mathbf{C} = \mathbf{I}$

In the first normalization, the entries of $\mathbf{C}$ are the correlations of the *weighted* criterion variables with the *weighted* object scores. In the second normalization, the distances between the rows of $\mathbf{Q}$ are approximated by the distances between the rows of $\mathbf{XA}$.

Normalization (a) is established by computing:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k(\mathbf{x}_k'\mathbf{V}\mathbf{x}_k)^{-1/2}\, n^{1/2}, \tag{40}$$

in case model is PCA. For RA $\mathbf{x}_k$ in (40) is replaced by $\mathbf{X}\mathbf{a}_k$. After having found the object scores in this way, the $\mathbf{C}$ is adjusted to keep the loss unchanged.

Normalization (b) is just a simple normalization of the columns of **C** with adjusting the **X** (in PCA) or the **A** (in RA) afterwards. When the model is CDA, we have a different normalization, which is on the within sums of squares; this has already been explained in section 3.3.

When the variables are weighted differently, there is no satisfying equivalent to normalization (a). One possibility is to normalize the object scores on $\mathbf{x}_k'\mathbf{V}^*\mathbf{x}_k$ where $\mathbf{V}^*$ represents the average weights matrix. This would incorporate the situation with equal weights as a special case. However, we have chosen to normalize the unweighted object scores to n, with the proper adjustment of **C** afterwards. For normalization (b) it does not matter if the weights are equal or different.

5. Computational Aspects

A straightforward possibility for constructing an algorithm, based on theory in the previous sections, is to alternate between the sets of parameters **Q**, **X**, **A** and **C** until convergence. In other words, for each set of parameters one update is computed, after which we continue with the next set of parameters. The decrease of the loss is evaluated at the end of a complete cycle through all sets.
In the present algorithm we have chosen for a different approach, which appears to be faster. Within the updating of the sets **X**, **A** and **C**, we iterate until convergence. Thus, after updating **X** for example, the decrease in loss is evaluated and we proceed with a next X-step, if this decrease in loss is larger than some criterion. Only after convergence of this *inner iteration loop*, we go on with updating the next set. The convergence criterion for these inner loops is not fixed, but chosen as a fraction of the difference of the two previous loss values computed after a complete cycle. For the criterion variables there is no inner loop, if we choose normalization (iii), since these are always optimally updated in one single step.

6. Discussion

In this paper, we have explained how four different type of models can be fitted to the data, by repeatedly solving a projection problem. From a technical point of view, the only differences between the models are in the normalization restrictions. The extension of the

usual least squares loss functions for these models with loss weights complicates the computation a lot. Furthermore, it should be realized that, when the weights are unequal to 1, the computed matrices may have different meanings than in the ordinary unweighted least squares. For instance, the matrix **C** doesn't represent the correlations anymore, for a general weights matrix. More practical experience has to learn us the merits of the MULTIPALS approach.

One important use of MULTIPALS is in iteratively reweighted least squares (IRLS) algorithms (Verboon, 1991), which involves solving a connected series of weighted least squares problems. In an IRLS algorithm the weights change in each major step; within each step, however, a complete MULTIPALS-step can be performed.

Another interesting application of MULTIPALS would be in modeling pairwise preference data by various vector models (Takane & Shibayama, 1991). Here the objects are pairs of options, and the pair comparison design is coded in **X**.

APPENDIX I

In this appendix, it will be shown why the normalization (ii), $\mathbf{q}_j'\mathbf{V}_j\mathbf{q}_j = n$, will be useless. Let $\mathbf{q}$ be any vector (we omit the index j for ease of notation) of a certain length to be specified implicitly later. Now define $\mathbf{q}^{(1)} = \mathbf{V}^{1/2}\mathbf{q}$ for any $\mathbf{V}$. This weighted variable is projected upon the cone of admissible transformations, yielding the restricted update $\mathbf{q}^{*(1)}$, which is assumed without loss of generality to have the required length, i.e. $\mathbf{q}^{*(1)'}\mathbf{q}^{*(1)} = n$. Now suppose we want this variable to have an additional weight α, which gives:

$$\mathbf{q}^{(2)} = \alpha^{1/2}\mathbf{V}^{1/2}\mathbf{q} = \alpha^{1/2}\mathbf{q}^{(1)}.$$

The vectors $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$ differ in length, but are in the same direction. It follows that their projections only differ in length, which is illustrated in Figure 1. So adjusting the length of both vectors to a constant (which is n) will always result in the same projected vector, independent of the weight factor α.

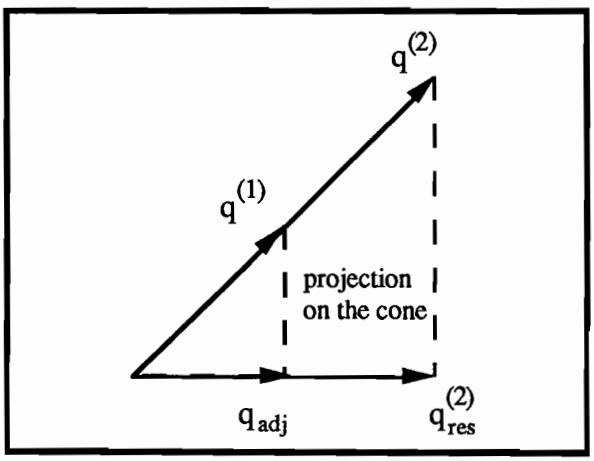


Figure 1. Illustration of the second normalization.

The sum of squares of the restricted (i.e. projected) variable $\mathbf{q}_{res}^{(2)}$ is:

$$\mathbf{q}_{res}^{(2)'}\mathbf{q}_{res}^{(2)} = \alpha^{1/2}\mathbf{q}^{*(1)'}\alpha^{1/2}\mathbf{q}^{*(1)} = \alpha n.$$

So to adjust $\mathbf{q}^{*(2)}$ to its required length, we compute

$$\mathbf{q}^{*(2)} = \alpha^{-1/2}\mathbf{q}_{res}^{(2)} = \mathbf{q}^{*(1)}.$$

So $\mathbf{q}^*$ is always the same for any choice of α, thus

$$\mathbf{q}_j^* = \alpha_j^{1/2}\mathbf{V}_j^{1/2}\mathbf{q}_j^{*(2)} = \mathbf{V}_j^{1/2}\mathbf{q}_j^{*(1)},$$

which yields for the unweighted criterion variables: $\mathbf{q}_j^{*(1)} = \alpha_j^{1/2}\mathbf{q}_j^{*(2)}$. Examining the loss function (see (8)),

$$\sigma(\mathbf{x}_k) = \sum_j(\mathbf{q}_j^* - \mathbf{x}_k c_{jk})'\mathbf{V}_j(\mathbf{q}_j^* - \mathbf{x}_k c_{jk}),$$

bringing the weight matrix within the brackets yields:

$$\sigma(\mathbf{x}_k) = \sum_j(\mathbf{V}_j^{1/2}\mathbf{q}_j^* - \mathbf{V}_j^{1/2}\mathbf{x}_k c_{jk})'(\mathbf{V}_j^{1/2}\mathbf{q}_j^* - \mathbf{V}_j^{1/2}\mathbf{x}_k c_{jk}),$$

we immediately notice that for any choice of α_j:

$$\mathbf{V}_j^{1/2}\mathbf{x}_k^{(1)}c_{jk}^{(1)} = \mathbf{V}_j^{1/2}\mathbf{x}_k^{(2)}c_{jk}^{(2)}\alpha_j^{1/2}.$$

So it follows that we may set $c_{jk}^{(1)} = c_{jk}^{(2)}\alpha_j^{1/2}$ and $\mathbf{x}_k^{(1)} = \mathbf{x}_k^{(2)}$ without altering the loss.

So multiplying a variable with a constant weight with normalization (ii) will not change $\mathbf{X}$, while $\mathbf{q}_j$ and $\mathbf{c}_j$ are only changed by a constant. It follows that such a normalization is rather useless in practice.

APPENDIX II

In this appendix, it is proved that a general problem of the form

$$\min_{\mathbf{x}\in\Omega} \sigma_{\mathbf{M}}(\mathbf{x}) = (\mathbf{y} - \mathbf{A}\mathbf{x})'\mathbf{M}(\mathbf{y} - \mathbf{A}\mathbf{x}), \tag{II.1}$$

which is called a generalized projection problem in the metric $\mathbf{M}$, can be decomposed in

$$\sigma_{\mathbf{M}}(\mathbf{x}) = \sigma_{\mathbf{M}}(\mathbf{x}^*) + (\mathbf{x}^* - \mathbf{x})'\mathbf{A}'\mathbf{M}\mathbf{A}\,(\mathbf{x}^* - \mathbf{x}), \tag{II.2}$$

where $\mathbf{x}^*$ is the unconstrained minimizer of $\sigma_{\mathbf{M}}(\mathbf{x})$ that satisfies

$$\mathbf{A}'\mathbf{M}\mathbf{A}\mathbf{x}^* = \mathbf{A}'\mathbf{M}\mathbf{y}. \tag{II.3}$$

Proof

Substituting the decomposition $\mathbf{x} = \mathbf{x}^* + (\mathbf{x} - \mathbf{x}^*)$ in (II.1) yields

$$\sigma_{\mathbf{M}}(\mathbf{x}) = \{(\mathbf{y} - \mathbf{A}\mathbf{x}^*) - \mathbf{A}(\mathbf{x} - \mathbf{x}^*)\}'\mathbf{M}\{(\mathbf{y} - \mathbf{A}\mathbf{x}^*) - \mathbf{A}(\mathbf{x} - \mathbf{x}^*)\}. \tag{II.4}$$

Expanding (II.4), it is clear that the first sum of squares term is equal to the first component of (II.2), the second sums of squares term is equal to the second component of (II.2), so it remains to be shown that the cross product term in (II.4) vanishes. Thus,

$$(\mathbf{x} - \mathbf{x}^*)'\mathbf{A}'\mathbf{M}(\mathbf{y} - \mathbf{A}\mathbf{x}^*) = \mathbf{x}'\mathbf{A}'\mathbf{M}\mathbf{y} - \mathbf{x}'\mathbf{A}'\mathbf{M}\mathbf{A}\mathbf{x}^* - \mathbf{x}^{*\prime}\mathbf{A}'\mathbf{M}\mathbf{y} + \mathbf{x}^{*\prime}\mathbf{A}'\mathbf{M}\mathbf{A}\mathbf{x}^* \tag{II.5}$$

should be equal to zero. Now, if $\mathbf{x}^*$ satisfies (II.3), then the first two terms in (II.5) cancel each other, as do the last two terms. Q.E.D.

Finally, if $\mathbf{x}$ is unconstrained, the second component of (II.2) - the only one involving $\mathbf{x}$ - can be made exactly zero by setting $\mathbf{x} = \mathbf{x}^*$, so in that situation we have $\sigma_{\mathbf{M}}(\mathbf{x}^*) = \sigma_{\mathbf{M}}(\mathbf{x})$.

APPENDIX III

In this appendix, we will prove that the matrix product $\mathbf{A'X'GD^{-1}G'XA}$, taken from the RA solution after convergence and rotation, is diagonal. We start by considering the estimated model from the loss function (1) after convergence and rotation:

$$\tilde{\mathbf{Q}} = \mathbf{XAC'}. \tag{III.1}$$

The matrix product $\mathbf{XA}$ is in principal axes orientation, so when we write down the singular value decomposition of $\tilde{\mathbf{Q}}$ as $\mathbf{K\Phi L'}$, we know that

$$\mathbf{XA} = \mathbf{KF}, \tag{III.2}$$

and

$$\mathbf{C} = \mathbf{L\Phi F^{-1}}, \tag{III.3}$$

with $\mathbf{F}$ a diagonal matrix, defining the appropriate normalization. The matrix $\mathbf{C}$ is also the least squares estimate for fixed $\mathbf{Q}$, $\mathbf{X}$ and $\mathbf{A}$, that is

$$\mathbf{C'} = (\mathbf{A'X'XA})^{-1}\mathbf{A'X'GD^{-1/2}}. \tag{III.4}$$

Now $\mathbf{C'C}$ can be written from (III.3) and (III.4) as

$$\mathbf{C'C} = \mathbf{F^{-1}\Phi^2F^{-1}} = \mathbf{F^{-2}A'X'GD^{-1}G'XAF^{-2}}, \tag{III.5}$$

and it follows that

$$\mathbf{A'X'GD^{-1}G'XA} = \mathbf{F\Phi^2F}, \tag{III.6}$$

which is diagonal. Q.E.D.

References

Gabriel, K.R. and Zamir, S. (1979). Lower rank approximations of matrices by least squares with any choice of weights. *Technometrics, 21, 4,* 489-498.

Gifi, A. (1990). *Nonlinear Multivariate Analysis*. New York: Wiley.

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

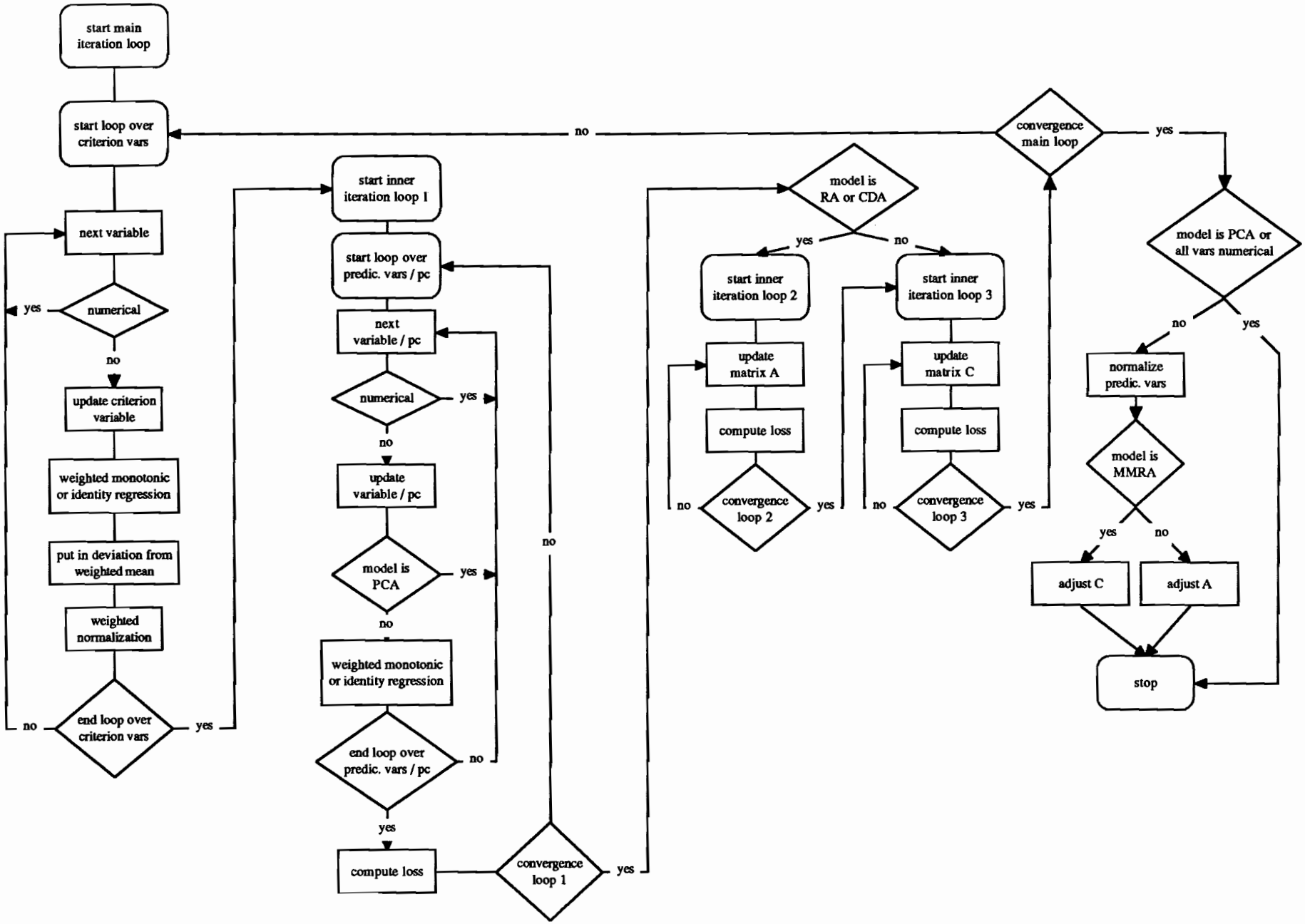
Heiser, W.J., Meulman, J.J., van der Lans, I.A. & van den Berg G.M. (1988). *Notes on MULTIPALS*. Research Report RR-88-09. Leiden: Department of Data Theory.

Takane, Y. and Shibayama, T. (1991). Principal component analysis with external information on both subjects and variables. *Psychometrika, 56,* 97-120.

Verboon, P. (1991). *Nonlinear Principal Component analysis: overview and new developments with respect to aspects of resistance*. Research Report RR-91-09. Leiden: Department of Data Theory.

Van de Geer, J.P. (1986). *Introduction to multivariate analysis*. Leiden: DSWO press.

Young, F.W., Takane, Y. & De Leeuw, J. (1978). The principal components of mixed measurement level multivariate data: an alternating least squares method with optimal scaling features. *Psychometrika, 43*, 279-281.


start main iteration loop
start loop over criterion vars
next variable
numerical
yes
no
update criterion variable
weighted monotonic or identity regression
put in deviation from weighted mean
weighted normalization
end loop over criterion vars
no
yes
start inner iteration loop 1
start loop over predic. vars / pc
next variable / pc
numerical
yes
no
update variable / pc
model is PCA
yes
no
weighted monotonic or identity regression
end loop over predic. vars / pc
no
yes
compute loss
convergence loop 1
no
yes
model is RA or CDA
yes
no
start inner iteration loop 2
update matrix A
compute loss
convergence loop 2
no
yes
start inner iteration loop 3
update matrix C
compute loss
convergence loop 3
no
yes
convergence main loop
no
yes
model is PCA or all vars numerical
no
yes
normalize predic. vars
model is MMRA
yes
no
adjust C
adjust A
stop