NOTES ON MULTIPALS

W.J. Heiser
J.J. Meulman
I.A. van der Lans
G.M. van den Berg

Department of Data Theory
University of Leiden
## Index

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1 On the algorithm to fit the Principal Components Model or the Multiple Regression Model</td>
<td>3</td>
</tr>
<tr>
<td>2 Treatment of Data Weights</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Breaking up projection into a number of successive projections.</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Fitting $Q$ for fixed $X$ &amp; $B$</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Column-conditional transformation of $Q$</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Row-conditional transformation of $Q$</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Fitting $X$ for fixed $Q$ &amp; $B$</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Point-wise fitting of $X$</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Dimension-wise fitting of $X$</td>
<td>9</td>
</tr>
<tr>
<td>3 MULTIPALS in the PATHALS framework</td>
<td>11</td>
</tr>
<tr>
<td>4 Redundancy analysis and MULTIPALS</td>
<td>14</td>
</tr>
<tr>
<td>4.1 Column- or row-wise fitting of $A$</td>
<td>15</td>
</tr>
<tr>
<td>4.2 Element-wise fitting of $A$</td>
<td>16</td>
</tr>
<tr>
<td>4.3 Normalization</td>
<td>17</td>
</tr>
<tr>
<td>5 Measurement level, normalization and rotation in MULTIPALS</td>
<td>17</td>
</tr>
<tr>
<td>5.1 Measurement level and normalization restriction</td>
<td>17</td>
</tr>
<tr>
<td>5.2 Rotation to principle components in PCA, RA and CDA</td>
<td>20</td>
</tr>
<tr>
<td>5.2.1 Rotation of the PCA Solution</td>
<td>21</td>
</tr>
<tr>
<td>5.2.2 Rotation of the RA Solution</td>
<td>21</td>
</tr>
<tr>
<td>5.2.3 Problems in the computation of rotated Solutions</td>
<td>23</td>
</tr>
<tr>
<td>6 Canonical discriminant analysis in MULTIPALS</td>
<td>23</td>
</tr>
<tr>
<td>6.1 The case of uniform weights</td>
<td>23</td>
</tr>
<tr>
<td>6.2 The case of non-uniform weights</td>
<td>27</td>
</tr>
<tr>
<td>References</td>
<td>29</td>
</tr>
<tr>
<td>Proposed syntax for MULTIPALS</td>
<td>30</td>
</tr>
</tbody>
</table>
Introduction

MULTIPALS is an acronym for MULTIple Projection analysis by Alternating Least Squares. It is more the name of a project than of an actual tested method or computer program. We started this project in an attempt to define a useful extension to the Gifi series HOMALS, PRINCALS, CANALS, etc., but unforseen circumstances caused considerable delays and set-backs.

In this report various separate memo's that circulated among the authors have been assembled, for easier future reference. Thus we do not claim completeness, consistency, or even a clear style. Yet we hope that the project will regain some of its original vitality some day in the near future.
1. On the algorithm to fit the Principal Components Model or the Multiple Regression Model

For both models we minimize the same loss function. It is written as

$$\sigma(Q;X;B) = \Sigma_{j} (q_{j} - \Sigma_{s} b_{js} x_{s})' (q_{j} - \Sigma_{s} b_{js} x_{s})$$

(1)

where $nQ = q_{1},...,q_{j},...,q_{m}$ denotes the (partially) known dependent variables,

$nX = x_{1},...,x_{s},...,x_{t},...,x_{p}$ denotes the independent variables, either unknown (PCA) or (partially) known (MR),

$mB = b_{1},...,b_{s},...,b_{t},...,b_{p}$ denotes the weights, or its transpose

$pB = b_{1},...,b_{j},...,b_{m}$ the loadings

Loss function (1) can be minimized by alternating three different steps. We assume that both the variables $q_{j}$ and the variables $x_{s}$ have sum of squares equal to 1. In the first step we find transformations of the $q_{j}$. This is done by first computing the unrestricted update, and performing monotone/identity regression on the result. The unrestricted update $q_{j}^*$ for given $X$ and $B$ is found as

$$q_{j}^* = Xb_{j}$$

(2)

For given $Q$ and $B$ we find the unrestricted update for $x_{t}$ as

$$x_{t}^* = Qb_{t} - \Sigma_{s \neq t} x_{s} b_{s} b_{t}.$$  

(3)

For the PCA model it suffices the normalize the result. For the Multiple Regression model, where $x_{t}$ is observed and treated ordinally or nominally, we perform monotone/identity regression again. Finally in the third step we compute $b_{t}^*$ for given $Q$ and $X$ as

$$b_{t}^* = Q' x_{t} - \Sigma_{s \neq t} b_{s} x_{s} x_{t}.$$  

(4)

Below a program will be described that incorporates these three steps. The input to the program that is common to both the PCA model and the MR model consists of the matrix $Q$. 

MULTIPALS(Q)

* SOME INITIAL OPERATIONS *

NORMALIZE Q AND COPY THE RESULT IN Q^0

CHOOSE MODEL: PCA or MR

IF MODEL = PCA
  - CHOOSE DIMENSIONALITY NDim
  - INITIALIZE X

IF MODEL = MR
  - READ PREDICTOR VARIABLES X
  - NORMALIZE X AND COPY THE RESULT IN X^0
  - NDim=NCOL(X)

INITIALIZE B

* THE ITERATION PROCESS *

* TRANSFORM THE VARIABLES in Q *

\[
\begin{align*}
&j = 0 \\
&j = j + 1 \\
&q_j = Xb_j \\
&q_j = \text{SCAL}(q_j, q_j) \\
&q_j = \text{NORM1}(q_j) \\
&\text{yes} \quad j < m \quad \text{?} \\
&\text{no} \\
\end{align*}
\]

* COMPUTE NEW VALUES FOR X; loop across dimensions *

\[
\begin{align*}
&Y = Q \\
&t = 0 \\
&t = t + 1 \\
&x_t = \text{UPDAT}(Y, X, B) \\
&\text{IF MODEL = MR:} \quad x_t = \text{SCAL}(x_t, x_t) \\
&x_t = \text{NORM1}(x_t) \\
&\text{yes} \quad t < \text{NDIM} \quad \text{?} \\
&\text{no} \\
\end{align*}
\]

* COMPUTE NEW VALUES FOR B; loop across dimensions *

\[
\begin{align*}
&Y = Q' \\
&t = 0 \\
&t = t + 1 \\
&b_t = \text{UPDAT}(Y, B, X) \\
&\text{yes} \quad t < \text{NDIM} \quad \text{?} \\
&\text{no} \\
\end{align*}
\]

* COMPUTE THE STRESS *

\[
\begin{align*}
&\text{DIF} = \text{STRESS}(Q, X, B) \\
&\text{yes} \quad \text{DIF} \geq \text{CRIT} \\
&\text{no} \\
\end{align*}
\]
IF MODEL = PCA:  

\[ X = \text{GRAM}(X) \]
\[ Y = Q' \]

\[ t = 0 \]
\[ b_t = \text{UPDAT}(Y, X, B) \]

\[ t = t + 1 \]

yes \[ t < \text{NDIM} \]?
no

DECOMPOSE \( B \) INTO \( B = KAL' \)

\[ B = BL \]
\[ X = XL \]

********************************************************************************
SUBROUTINE \text{UPDAT}(Y, U, V)
********************************************************************************

* COMPUTES NEW VALUES FOR BOTH \( x_t \) AND \( b_t \) *

\[ u_t = Yv_t - \Sigma_{s \in t} u_s v_s v_t \]

********************************************************************************

The algorithm described above has been programmed in APL. The next step will involve restrictions on the \( B \) matrix, in order to obtain reduced rank regression. Here \( X \) is a given \( nxm_1 \) matrix and \( Q \) is an \( nxm_2 \) matrix. Then \( B \) must satisfy \( B = AC' \), where \( A \) is \( m_1 \times p \) and \( C' \) is \( pxm_2 \), with \( p \leq m \) where \( m = \text{minimum}(m_2, m_1) \). This restriction can be implemented by using \text{UPDAT} again.

2. Treatment of data weights.

This section describes the major algorithmic features of \textsc{multipals}, a method and program for \textsc{multiple} Projection analysis by Alternating Least Squares. Reasons for developing \textsc{multipals}. Comparison roughly with \textsc{princals}, \textsc{transreg}, \textsc{morals}, etc. The \textsc{multipals} loss function is:

\[ \sigma^2(Q, X, B) = \Sigma_{i} \Sigma_{j} w_{ij} (q_{ij} - \Sigma_{k} x_{ik}b_{jk})^2, \quad (5) \]

where \( Q = \{q_{ij}\} \) is an \( nxm \) matrix of scores, or \textit{quantifications}, for \textit{n} objects on \textit{m} response variables; \( X = \{x_{ik}\} \) is an \( nxp \) matrix of scores for the same analysis objects on \textit{p} predictor variables, \( B = \{b_{jk}\} \) is an \( nxp \) matrix of coefficients, or \textit{loadings}, describing the effect of the \textit{k}th predictor variable on the \textit{j}th response variable; \( W = \{w_{ij}\} \) is an \( nxm \) matrix of data weights, included for enhanced flexibility (examples to be explained later).
The two major special cases are: \( X \) (partially) known, various forms of multiple regression and additivity analysis; \( X \) (partially) unknown, various forms of component analysis. \( Q \) is always (partially) known, \( B \) might be constrained (examples to be explained later).

In matrix notation (5) can be written as a sum of squared Euclidean norms across rows, or as a sum of squared Euclidean norms across columns. Thus we find

\[
\sigma^2(Q,X,B) = \sum_i \| q_i - Bx_i \|^2_{U(i)} , \tag{6a}
\]

where the squared weighted norm in \( n \)-dimensional space

\[
\| q_i - Bx_i \|^2_{U(i)} = (q_i - Bx_i)'U(i)(q_i - Bx_i) \tag{6b}
\]

is the basic constituent; or, alternatively, (5) is equivalent to

\[
\sigma^2(Q,X,B) = \sum_j \| q_j - Xb_j \|^2_{V(j)} , \tag{7a}
\]

in which the squared weighted norm in \( m \)-dimensional space

\[
\| q_j - Xb_j \|^2_{V(j)} = (q_j - Xb_j)'V(j)(q_j - Xb_j) \tag{7b}
\]

is used. In (6b) and (7b) \( q_i \) and \( q_j \) are the rows and columns of \( Q \), respectively, written as \( m(n) \)-vectors; similarly, the \( p \)-vector \( x_i \) denotes the \( i \)th row of \( X \) and the \( p \)-vector \( b_j \) the \( j \)th row of \( B \); the notation \( U(i) \) is used for a \( mxm \) diagonal matrix containing the elements of the \( i \)th row of \( W \) on the diagonal, and \( V(j) \) for a \( nxn \) diagonal matrix containing the elements of the \( j \)th column of \( W \) on the diagonal.

In the MULTIPALS loss function \( Q, X, B, \) and sometimes \( W \) are all variable under certain constraints (explain), so the general strategy will be ALS (explain). For any fixed \( y \) and \( A \) a problem of the form

\[
\min_{z \in \Omega} \eta_2^M(z) = \| y - Az \|^2_M , \tag{8}
\]

is called a (generalized) projection problem in the metric \( M \). It is a simple projection problem if \( z \) is unconstrained, and \( M = I \), the identity. The classical solution of the simple case is obtained for \( y^* = Az^* = (A'A)^{-1}A'y \), where \( A(A'A)^{-1}A' \) is called the projection matrix, or simply the
projector. The problem considered here is generalized in the sense that (a) \( z \) can be constrained in various ways, as indicated in the notation \( z \in \Omega \), and (b) the matrix \( M \) is any positive semi-definite (p.s.d.) matrix. The name MULTIPALS derives from the fact that (5) can be minimized by repeatedly solving a number of problems that are all of the form (8).

2.1. Breaking up projection into a number of successive projections.
A further major idea of algorithm construction, in the tradition of the Gifi system, is that complicated projection problems can always be solved by cycling through a series of simpler ones, so that in general we don't actually use projection matrices unless they have a particularly convenient form. Also, simplifications can frequently be achieved by using the additive decomposition of \( \eta^2_M(z) \) obtained when one substitutes the particular decomposition of its argument \( z = z^* + (z - z^*) \), where \( z^* \) is the unconstrained minimum of \( \eta^2_M(z) \). In general one gets

\[
\eta^2_M(z) = \eta^2_M(z^*) + \| z^* - z \|^2_{A'MA},
\]

only the last part of which involves \( z \). An elementary proof of this runs as follows.

**Proof**
Suppose \( z^* \) satisfies

\[
A'MAz^* = A'My;
\]

using this \( z^* \) in the above-mentioned decomposition of \( z \), the squared weighted norm becomes

\[
\eta^2_M(z) = \langle y - Az \; \rangle M (y - Az)
= \{ (y - Az^*) - A(z - z^*) \} M \{ (y - Az^*) - A(z - z^*) \}.
\]

Expanding (11) it is clear that its first sum of squares term is equal to the first component of (9), its second one is equal to the second component of (9), so there remains to be shown that the cross product term in (11) vanishes. We find

\[
(Az - Az^*)M(y - Az^*) = z'A'My - z'A'MAz^* - z^*A'My + z^*A'MAz^*.
\]

Now if \( z^* \) satisfies (10), then the first two terms in (12) cancel each other, as do the last two terms. Finally, if \( z \) is unconstrained, the second component of (9) - the only one involving \( z \) - can be made exactly equal to zero by actually choosing \( z^* \); so \( z^* \) is the unconstrained minimizer of \( \eta^2_M(z) \). ∴
One of the consequences of being able to decompose $\eta^2_M(z)$ in this fashion is the flexibility it provides for bringing a variety of problems into one of a number of basic forms. This will become clear in the remainder of this report.

### 2.2. Fitting $Q$ for fixed $X$ and $B$.

Symbols printed in outline, such as $X$ and $B$, will be used whenever we regard the ALS subproblem of minimizing the loss function with respect to the remaining parameters in which the outlined parameters are held fixed at their current value. In MULTIPALS two cases are distinguished for treating $Q$, called column-conditional and row-conditional transformation.

#### 2.2.1. Column-conditional transformation of $Q$.

This is the more or less standard case, as most data files are conventionally coded with 'cases' or 'units of analysis' as rows and 'variables' as columns. The most convenient expression of the loss function now is (7a), since it provides a natural split into $m$ components of the form

$$
\sigma^2(q_j, X, B) = \| q_j - q_{ij} \|_2 V(j),
$$

where $q_{ij} = Xx_j$ is a fixed $n$-vector. Depending on the measurement level associated with column $j$, (13) is an intra-class regression (when LEVEL=NOMINAL), an isotonic regression (when LEVEL=ORDINAL), a spline regression (when LEVEL=SPLINE), or a linear regression problem (when LEVEL=NUMERICAL), optionally with weights $V(j)$.

#### 2.2.2. Row-conditional transformation of $Q$.

This is the less standard case, useful when the data is coded in the reverse fashion, as it frequently is in situations that call for a 'reversed indicator matrix' in the Gifi system (explain). The most convenient expression of the loss function now becomes (6a), since it provides a natural split into $n$ components of the form

$$
\sigma^2(q_i, X, B) = \| q_i - q_{ij} \|_2 U(i),
$$

where $q_{ij} = Bx_i$ is a fixed $m$-vector. Depending on the measurement level associated with row $i$, (14) is one of the regression problems discussed in 2.1, optionally with weights $U(i)$ this time, and of order $m$ rather than $n$.

### 2.3. Fitting $X$ for fixed $Q$ and $B$.

For fixed $Q$ and $B$ the row- and column-wise split of the loss function is useful for fitting either separate points or separate dimensions. In MULTIPALS point-wise fitting of $X$ is in the present
design called for only when supplementary, or passive, objects have to be scaled, while
dimension-wise fitting of $X$ is the strategy used during iteration. If there are (measurement)
constraints on $X$, it is always assumed here that they are dimension-wise (but we could drop this
restriction of flexibility).

2.3.1. **Point-wise fitting of $X$.**

From (6a) it follows that this involves successively solving $n$ problems of type

$$
\min_{x_i} \| q_{ii} - B x_i \|_{U(i)}^2 .
$$

(15)

From the general theory it follows immediately that $x_i^*$ is a solution to (15) when it satisfies

$$
B^T U(i) B x_i^* = B^T U(i) q_{ii} ;
$$

(16)

which implies that we have to solve a system of $p$ linear equations. Notice that in the principal
components situation scaling a supplementary object means that $q_{ii}$ must contain the category
quantifications applying to object $i$ that were obtained in the analysis of the other objects. In the
multiple regression situation we also have $x_i$, the category quantifications applying to object $i$ in
the predictor set, so that (16) is not needed, but $q_{ii}$ and $B x_i$ can be compared in a measure of
**prediction error**, or **cross-validation**, built up from components of the form (15).

2.3.2. **Dimension-wise fitting of $X$.**

This is the type of fitting used earlier in **CANALS** and **OVERALS**, and in Meulman and Heiser
(1984), although due to the presence of weights the formulation here is more general. It is very
similar to the so-called reduced rank individual differences scaling case in **PROXSCAL** (Heiser and
Stoop, 1986). The most convenient expression of the loss function now again becomes (7a),
although the natural split into $m$ components is not immediately helpful. In order to fit dimension
$k$ while the other coordinates remain fixed, let us first define the dimension-wise split of
coordinates

$$
X = x_1 e_1' + ... + x_k e_k' + ... + x_p e_p' ,
$$

(17)

where $e_k$ denotes the $k$th column of the identity matrix. Then the auxiliary configuration

$$
X_{(-k)} = X - x_k e_k' ,
$$

(18)
i.e. the configuration obtained from \( \mathbf{X} \) by annihilating the coordinates of the \( k \)th dimension, allows us to write

\[
\begin{align*}
\mathbf{X} &= \mathbf{X}_{(-k)} + \mathbf{x}_k \mathbf{e}_k', \\
\mathbf{q}_j - \mathbf{X}_{\mathbb{B}_j} &= \mathbf{q}_j^0 - \mathbf{x}_k \mathbf{e}_k' \mathbb{B}_j, \\
\mathbf{q}_j^0 &= \mathbf{q}_j - \mathbf{X}_{(-k)} \mathbb{B}_j.
\end{align*}
\]  

(19a) (19b) (19c)

The notation \( \mathbf{X}_{(-k)} \) used in (19a) and (19c) indicates that we may consider it as a fixed matrix once we concentrate on fitting the \( k \)th dimension. Using these definitions the MULTIPALS loss function (7a) can be written as

\[
\sigma^2(\mathbb{Q}, \mathbf{x}_k, \mathbb{B}) = \sum_j \| \mathbf{q}_j^0 - \mathbf{x}_k \mathbf{e}_k' \mathbb{B}_j \|^2 v(j).
\]  

(20)

In order to simplify this, first notice that we may write more simply \( \mathbf{e}_k' \mathbb{B}_j = \mathbf{b}_{jk} \), the \( jk \)th element of the fixed matrix \( \mathbb{B} \), since \( \mathbf{e}_k \) selects the \( k \)th loading of variable \( j \) in \( \mathbb{B}_j \). We now use the general rule described in the introduction. The unconstrained minimum of (20) is attained when \( \mathbf{x}_k^* \) satisfies

\[
\mathbf{V}^*(k) \mathbf{x}_k^* = \sum_j \mathbf{b}_{jk} \mathbf{v}(j) \mathbf{q}_j^0,
\]  

(21)

where the aggregated weight matrix \( \mathbf{V}^*(k) \) is defined as

\[
\mathbf{V}^*(k) = \sum_j \mathbf{b}_{jk}^2 \mathbf{v}(j).
\]  

(22)

At the right-hand side of (21) the corrected variable quantifications \( \mathbf{q}_j^0 \) are weighted by \( \mathbf{v}(j) \) to the extent of their component loading \( \mathbf{b}_{jk} \) on dimension \( k \), and in forming \( \mathbf{V}^*(k) \) the variable-wise weights are aggregated to the extent of the square of their component loadings. Loss function (20) can now be decomposed as

\[
\sigma^2(\mathbb{Q}, \mathbf{x}_k, \mathbb{B}) = \sigma^2(\mathbb{Q}, \mathbf{x}_k^*, \mathbb{B}) + \| \mathbf{x}_k - \mathbf{x}_k^* \|^2 v^*_k.
\]  

(23)

So in the principal component situation, where \( \mathbf{X} \) is unconstrained, it is enough to solve the simple system of \( n \) linear equations (21), for each \( k \). These systems are simple because \( \mathbf{V}^*(k) \) is diagonal. In the multiple regression situation, where \( \mathbf{X} \) is constrained by the measurement restrictions on the predictor variables, (23) shows that in addition we have to solve a (intra-class, isotonic, spline, or linear) regression problem of projecting \( \mathbf{x}_k^* \) in the metric \( \mathbf{V}^*(k) \).

Two remarks of some practical importance are in order. In the first place, although each step (i.e., finding a new \( \mathbf{x}_k \)) is made so that the MULTIPALS loss function decreases, and therefore
could be made only once (for fixed Q and B), the real minimum of \( \sigma^2(Q;X;B) \) is attained only if those steps are repeated until convergence, since every one of them is conditional upon the current values of the other dimensions. So it is useful to keep the matrix \( \sum_j V(j)_q \beta_j R(j)_q \) stored, which is constant throughout these iterations, and to select the \( k \)th column of it each time it is needed (at the right-hand side of (21)). In practice we may want to stop somewhere between one cycle and complete convergence, depending on the behaviour of the master algorithm.

Secondly, notice how (22) simplifies when \( V(j) = V \) for all \( j \). We find \( \mathbf{V}^*(k) = \beta_k \mathbf{V} \), with \( \beta_k \) the \( k \)th diagonal element of the matrix \( \mathbf{B}^T \mathbf{B} \); since a scalar quantity does not alter the argument for which a minimum is attained, solutions of (23) for different dimensions can all be computed with the same matrix \( \mathbf{V} \) (of course, if (23) is used to determine fit values, \( \beta_k \) should be included in the summation).

3. MULTIPALS in the PATHALS framework

We start with the ordinary PCA loss function

\[
\sigma(Q;X;B) = \sum_j (q_j - \Sigma_s b_{js}x_s)'(q_j - \Sigma_s b_{js}x_s),
\]  

(24)

and we remember from Section 1 that

\[
q_j^* = Xb_j
\]  

(25)

\[
x_t^* = Qb_t - \Sigma_{s \neq t} x_s b_s b_t,
\]  

(26)

\[
b_t^* = Q'x_t - \Sigma_{s \neq t} b_s x_s x_t
\]  

(27)

In this section we will inspect the PATHALS approach to Principal Components Analysis, and relate it to the MULTIPALS approach. As an example we will take a Principal Components model with five manifest variables and two latent variables. Then there are five equations to be fitted:

\[
q_1 = b_{11}x_1 + b_{12}x_2
\]  

(28a)

\[
q_2 = b_{21}x_1 + b_{22}x_2
\]  

(28b)

\[
q_3 = b_{31}x_1 + b_{32}x_2
\]  

(28c)

\[
q_4 = b_{41}x_1 + b_{42}x_2
\]  

(28d)

\[
q_5 = b_{51}x_1 + b_{52}x_2
\]  

(28e)

In the PATHALS framework these equations are described in a Mxp matrix \( \mathbf{A} \), where \( M \) denotes the total number of (manifest and latent) variables, and \( p \) denotes the number of different
equations (in our case p equals m, the number of manifest variables). For the example the matrix \( A \) is given by the following Table.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>.</th>
<th>s</th>
<th>.</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>k</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>m</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>.</td>
<td>-b11</td>
<td>-b21</td>
<td>-b31</td>
<td>-b41</td>
<td>-b51</td>
</tr>
<tr>
<td>M</td>
<td>-b12</td>
<td>-b22</td>
<td>-b32</td>
<td>-b42</td>
<td>-b52</td>
</tr>
</tbody>
</table>

The idea is as follows. \( Y \) denotes the complete set of variables \( \{Q, X\} \). Then:

- when variable \( y_j \) is the dependent variable in equation \( s \) then \( a_{ks} = 1 \);
- when variable \( y_j \) is an independent variable in equation \( s \) then \( a_{ks} = -b_{js} \);
- when variable \( y_j \) is not involved in equation \( s \) then \( a_{ks} = 0 \).

Loss function (24) can now be rewritten as

\[
\sigma(Y; A) = \text{tr} A'Y'YA, \tag{29}
\]

which should be minimized over \( Y \) and over \( A \), where some of the elements of \( A \) (that are equal to 1 or 0) are fixed. At the moment it is not immediately obvious how to minimize (29). Instead, we would like to have a loss function that compares each of the variables in \( Y \) with a weighted sum of all the other variables. There is a way to do this, but it is unclear whether this makes sense. First let us have a look at the matrix \( AA' \) in the following Table.
The matrix $AA'$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>k</th>
<th>m</th>
<th></th>
<th></th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0 0</td>
<td>0 0</td>
<td>-b(_{11})</td>
<td>-b(_{12})</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1 0</td>
<td>0 0</td>
<td>-b(_{21})</td>
<td>-b(_{22})</td>
<td></td>
</tr>
<tr>
<td>k</td>
<td>0</td>
<td>0 1</td>
<td>0 0</td>
<td>-b(_{31})</td>
<td>-b(_{32})</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0 1</td>
<td>0 0</td>
<td>-b(_{41})</td>
<td>-b(_{42})</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>0</td>
<td>0 0</td>
<td>0 1</td>
<td>-b(_{51})</td>
<td>-b(_{52})</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-b(_{11})</td>
<td>-b(_{21})</td>
<td>-b(_{31})</td>
<td>-b(_{41})</td>
<td>-b(_{51})</td>
<td>b(<em>{11})b(</em>{12})+...+b(<em>{31})b(</em>{32})+...+b(<em>{51})b(</em>{52})</td>
</tr>
<tr>
<td>M</td>
<td>-b(_{12})</td>
<td>-b(_{22})</td>
<td>-b(_{32})</td>
<td>-b(_{42})</td>
<td>-b(_{52})</td>
<td>b(<em>{11})b(</em>{12})+...+b(<em>{31})b(</em>{32})+...+b(<em>{51})b(</em>{52})</td>
</tr>
</tbody>
</table>

Now we define

$$C = -(AA'),$$

and write the loss function as

$$\sigma(Y; C) = \sum_k (y_k - \sum_{l \neq k} c_{lk} y_l)'(y_k - \sum_{l \neq k} c_{lk} y_l) \text{ for } k, l = 1, \ldots, M.$$  

[Loss function (31) can be found in Coolen and De Leeuw (1987, page 10), but $C$ is nowhere defined. They just state that "every variable is expressed as a linear combination of those variables with which it is connected through an arrow in the path diagram"].

From (31) the unrestricted $y_k^*$, whether it is manifest or latent, for given $y_l$ is found as

$$y_k^* = \sum_{l \neq k} c_{lk} y_l = Yc_k - y_k c_{kk}$$

Let us have a look at the matrices we are working with.
Some exercises to see whether (32) holds for the PCA model. For the manifest variable $q_1^*$ we have

\[ q_1^* = b_{11}x_1 + b_{12}x_2, \text{ or more general} \]

\[ q_j^* = Xb_j \tag{33} \]

For the latent variable $x_1^*$ we have

\[ x_1^* = b_{11}q_1 + b_{21}q_2 + b_{31}q_3 + b_{41}q_4 + b_{51}q_5 - (b_{11}b_{12} + \ldots + b_{31}b_{32} + \ldots + b_{51}b_{52})x_2 \tag{34a} \]

or more general

\[ x_t^* = Qb_t - \Sigma_{s=1}^t x_s b_s b_t \tag{34b} \]

so we are back at (25) and (26).

4. Redundancy Analysis and MULTIPALS

The least squares loss function of redundancy analysis can be written as follows (e.g. Van de Burg, 1988):
\[ \sigma^2(Q,X,A,C) = \text{tr} (Q - XAC')(Q - XAC') \]  
(35)

where \( A \) is a \( p \times r \) matrix and \( C \) is a \( m \times r \) matrix, and \( r \leq \min(p,m) \).

With the introduction of the matrix of data weights \( W \), the loss function becomes

\[ \sigma^2(Q,X,A,C) = \Sigma_j \left( q_j^i - XAc_j \right)' V_j \left( q_j^i - XAc_j \right) \]  
(36)

with \( c_j \) the \( j \)th column vector of \( C' \)

or, equivalently,

\[ \sigma^2(Q,X,A,C) = \Sigma_i \left( q_i^l - CA'x_i \right)' U_i \left( q_i^l - CA'x_i \right). \]  
(37)

Solving for \( Q, X, \) and \( C \) successively with other parameters remaining fixed poses the same problems as treated in Section 2. The only difference lies in minimizing the loss function over \( A \).

### 4.1 Column- or row-wise fitting of \( A \).

Define

\[ A_{(-l)} = A - a_l e_l', \]  
(38a)

with \( e_l \) the \( l \)th column of the \( r \times r \) identity matrix, and \( a_l \) the \( l \)th column vector of \( A \),

\[ q_j^0 = q_j^i - XA_{(-l)}c_j, \]  
(38b)

then to fit \( a_l \), the following loss function should be minimized:

\[ \sigma^2(a_l) = \Sigma_j \left( q_j^0 - Xa_l e_l' c_j \right)' V_j \left( q_j^0 - Xa_l e_l' c_j \right) \]  
(39)

which is equal to:

\[ \sigma^2(a_l) = \Sigma_j \left( q_j^0 - Xa_l c_{jl} \right)' V_j \left( q_j^0 - Xa_l c_{jl} \right). \]  
(40)

Equation (40) is minimized when we take:

\[ \Sigma_j (X c_{jl})' V_j (X c_{jl}) a_l = \Sigma_j (X c_{jl})' V_j q_j^0. \]  
(41)
However, in general $\Sigma_j (c_{ji}) V_j (c_{ji})$ will not be a diagonal matrix and therefore computation of its inverse might be time-consuming.

In complete analogy, we can minimize (37) by cycling through successive rows of $A, a_k$. In that case we get $p$ sets of $r$ linear equations:

$$
\Sigma_i (C x_{ik}) U_i (C x_{ik}) a_k = \Sigma_i (C x_{ik})^T U_i \varphi_i^0.
$$

(42)

Here $\Sigma_i (C x_{ik}) U_i (C x_{ik})$ will, in general, not be diagonal.

4.2 Element-wise fitting of $A$.

Define

$$
\bar{A}_{-kl} = A - e_k a_{kl} e_l^T
$$

(43a)

with $e_k$ the $k$th column of the $p \times p$ identity matrix and $e_l$ the $l$th column of the $r \times r$ identity matrix,

$$
\varphi_j^0 = \varphi_j - \bar{X} A_{-kl} e_j^T,
$$

(43b)

then to fit $a_{kl}$, the following loss function should be minimized:

$$
\sigma^2(a_{kl}) = \Sigma_j (\varphi_j^0 - \bar{X} e_k a_{kl} e_l^T e_j^T V_j (\varphi_j^0 - \bar{X} e_k a_{kl} e_l^T e_j^T),
$$

(44)

which is equal to:

$$
\sigma^2(a_{kl}) = \Sigma_j (\varphi_j^0 - \bar{X} e_k a_{kl} c_{ jl}^T V_j (\varphi_j^0 - \bar{X} e_k a_{kl} c_{ jl}^T).
$$

(45)

Equation (11) is minimized when we take:

$$
\Sigma_j (x_k c_{jl}) V_j (x_k c_{jl}) a_{kl} = \Sigma_j (x_k c_{jl}) V_j \varphi_j^0.
$$

(46)

This solution to (46) is simple since $\Sigma_j (x_k c_{jl}) V_j (x_k c_{jl})$ and $\Sigma_j (x_k c_{jl}) V_j \varphi_j^0$ are scalars. Thus we can cycle through elements of $A$. During this procedure the $(p \times r)$ values $\Sigma_j (x_k c_{jl}) V_j (x_k c_{jl})$ remain constant and need only to be computed once. The element-wise approach has been implemented in the current APL version of MULTIPALS.
4.3 Normalization

In redundancy analysis the matrices A and C are determined up to a rotation, which leaves AC' invariant. Therefore normalization restrictions are necessary to get an unique solution. A quite common normalization is that \( A'X'XA/n = I \) (Van der Burg, 1988). It is not necessary to normalize during iterations. Both A and C can be rotated after convergence of the algorithm. Here, the procedure for rotation to principal axes in PCA given in Section 1 is useful. If we replace X by XA and B by C' in this procedure, the rotation matrix L is computed. Then, the normalized solution is obtained by C'=CL and A=AL. However, the procedure given earlier is only valid in case all elements of W are equal. A generalized procedure for rotation will be given further on.

5. Measurement level and normalization and rotation in MULTIPALS

5.1 Measurement level and normalization restrictions.

The MULTIPALS loss function can be written as (cf (36)):

\[
\sigma^2(Q,X,A,C) = \sum_j (q_j - XAe_j)'V(j)(q_j - XAe_j)
\]  

(47)

where Q is an n x m matrix of scores of n objects on m response variables, X is an n x p matrix of the n objects on p predictor variables, A is a p x r matrix and C is a m x r matrix, and e_j is the j th column of C'.

To obtain the least squares loss functions of principal components analysis (PCA) or multivariate multiple regression analysis (MMRA), we define \( r = p \) and \( A=I \). To obtain the least squares loss functions of redundancy analysis (RA) or canonical discriminant analysis (CDA), we define \( r \leq \min(p,m) \).

The alternating least squares procedure to compute the solution to equation (47) iterates through a number of subproblems. Each of these subproblems involves the computation of optimal estimates for one of the sets of parameters Q, X, A or C at a time, with remaining sets of parameters held fixed. Ways to find unrestricted optimal estimates of Q, X, A and C are given in Section 2 and Section 4. However, in many instances the estimates for Q, X, A and C are restricted by measurement level and/or normalization restriction. The general procedure to find the restricted optimal estimates is to compute unrestricted optimal estimates, after which these unrestricted optimal estimates are projected upon the cone defined by measurement level and/or normalization restrictions, in the appropriate metric V*.
The specific restrictions upon each set of parameters are dictated by the kind of analysis to be performed with MULTIPALS. In MMRA each column of \( Q \) and \( X \) is restricted by its measurement level and to have zero mean and unit variance. In addition, there are no restrictions upon \( C \).

In PCA each column of \( Q \) is restricted by its measurement level and to have zero mean and unit variance. Each column of \( X \) is restricted to have zero mean and in addition it is required that \( X'X/n = I \). No restrictions are imposed upon \( C \).

In RA each column of \( Q \) and \( X \) is restricted by its measurement level and to have zero mean and unit variance. Moreover, we can choose between either the first normalization condition with the restriction, \( A'X'XA/n = I \), or the second normalization condition with the restriction, \( C'C = I \). In CDA each column of \( X \) is restricted by its measurement level and to have zero mean and unit variance. Each column of \( Q \) is restricted to have zero mean and in addition \( QQ = I \). Further, it is required that \( C'C = I \).

In case all diagonal elements of \( V^* \) are equal to each other, the restricted optimal estimates can be obtained by successive projections of the unrestricted optimal estimates onto the closed convex cones defined by respective restrictions with regard to measurement level, restrictions with regard to the mean and restrictions with regard to orthogonality restrictions. When there is also a restriction upon the sum of squares of the optimal estimates, the estimates can be normalized afterwards (De Leeuw, 1977a). Because all diagonal elements of \( V^* \) are equal, it can be shown that the projection onto the next cone will remain within the previous cone. This can be proven either directly (e.g. numerical measurement level) or indirectly by proving that the null spaces of the projections (Van de Geer, 1986) are orthogonal\(^1\) to each other and therefore the projection onto the next cone remains orthogonal to the antiprojection on the previous cone (e.g. nominal measurement level). However, when the diagonal elements of \( V^* \) are not equal to each other this property does not hold. Moreover, the results on normalized cone regression given by the Leeuw (1977a) do not hold anymore.

The general procedure used in MULTIPALS to compute restricted optimal estimates in the metric defined by any diagonal matrix \( V^* \) is *iterative majorization by quadratics* (see for instance, De Leeuw, 1977b; Dempster, Laird & Rubin, 1977; Heiser, 1987a). The subproblem of the computation of restricted optimal estimates, given the unrestricted optimal estimates, can be written as the minimization of:

\[ \min_{Y} \langle V^*, Y \rangle \text{ subject to } A'Y = b, \]

\[ \text{subject to } C'C = I. \]

\[^{1}\text{This is not true when consider a cone defined by orthogonality restrictions and a cone defined by measurement level restrictions. However these two kinds of restrictions never apply to the same set of estimates.} \]
\[ \xi(z) = (z - z^*)(V^*(z - z^*) \) \]
\[ z \in \Gamma \]

where
- \( z \) is the vector of restricted estimates,
- \( \Gamma \) is the cone defined by all restrictions upon \( z \),
- \( z^* \) is the vector of unrestricted estimates,
- \( V^* \) is a diagonal weight matrix.

Majorization relies on the fact that the function

\[ \xi(z, z) = \xi(z) + \beta (z - z)'(z - z) - 2(z - z)'V^*(z^* - z) \]

where
- \( z \) is an element of \( \Gamma \), and is called a supporting point,
- \( \beta \) is the largest eigenvalue of \( V^* \), which is equal to the largest element of \( V^* \) since \( V^* \) is a diagonal matrix,

is never smaller than \( \xi(z) \) and that \( \xi(z, z) = \xi(z) \).

After regrouping it can be found that

\[ \xi(z, z) = \alpha + \beta (z^\wedge - z)(z^\wedge - z) \]

where \( \alpha \) denotes a constant term, and

\[ z^\wedge = z + (1/\beta) V^*(z^* - z) . \]

Thus starting with \( z \in \Gamma \), we can compute an update of \( z \in \Gamma \) for the current value of \( z \) by projection of \( z^\wedge \) upon \( \Gamma \). Since this is a projection in the metric defined by \( I \), successive projections and results of normalized cone regression can be used. By iterating this step, while replacing \( z \) by the last estimate of \( z \) each time, the loss function is guaranteed to become smaller after each step. Iteration stops when the decrease in loss is less than some convenient criterion.

It can be noticed that, in case all diagonal elements of \( V^* \) are equal, \( (1/\beta) V^* \) is equal to \( I \) and therefore \( z^\wedge \) is equal to \( z^* \). As a consequence the majorization will take two iterations (one in which optimal restricted estimates are computed and one in which nothing changes because \( z \) contains the optimal restricted estimates and therefore the decrease in loss is exactly zero).
5.2 Rotation to principal components in PCA, RA and CDA\(^2\).

Suppose a solution is found that minimizes equation (47) and that the optimal estimates for \(Q, X, A\) and \(C\) are contained in respectively \(Q^{*}, X^{*}, A^{*}\) and \(C^{*}\). Now, for PCA we want to find the rotation matrix \(R\) such that \(X = X^{*}R'\) and \(C = C^{*}R'\) minimizes

\[
\sigma_t^2(X, C) = \sum_j \left( q^{*j} - X_t \cdot c_{jt} \right)'V(j) \left( q^{*j} - X_t \cdot c_{jt} \right) \tag{51}
\]

where
- \(X_t\) is a matrix containing the first \(t\) columns of \(X\),
- \(c_{jt}\) is a vector containing the first \(t\) elements of the \(j\)th column of \(C'\),
- \(x_t\) is the \(t\)th column of \(X\).

for consecutive values of \(t \leq p\), and under the restriction that \(u'x_t = 0, x_t'x_s = 0\) for each \(s < t\), and \(x_t'x_t = n\).

In RA the problem is to find a transformation matrix \(R\) such that \(A = A^{*}R'\) and \(C = C^{*}R'\) minimizes

\[
\sigma^2(A, C) = \sum_j \left( q^{*j} - X^*A_t \cdot c_{jt} \right)'V(j) \left( q^{*j} - X^*A_t \cdot c_{jt} \right) \tag{52}
\]

where
- \(A_t\) is a matrix containing the first \(t\) columns of \(A\),
- \(c_{jt}\) is a vector containing the first \(t\) elements of the \(j\)th column of \(C'\),
- \(a_t\) is the \(t\)th column of \(A\),

for each value of \(t \leq p\). Furthermore there are two normalization conditions. The first normalization condition requires that \(a_t'X^*X^*a_s = 0\) for each \(s < t\), and \(a_t'X^*X^*a_t = n\). The second normalization condition requires that \(c_t'c_s = 0\) for each \(s < t\), and \(c_t'c_t = 1\), where \(c_t\) is the \(t\)th column of \(C\).

The rotated solutions have the desirable property that the dimensions are nested given the optimal transformations (\(Q^*\) in PCA, or \(Q^*\) and \(X^*\) in RA). For example, the first dimension of a two dimensional PCA of \(Q^*\) is equal to the one dimensional PCA of \(Q^*\). The first two dimensions of a three dimensional PCA of \(Q^*\) are equal to the dimensions of the two dimensional PCA of \(Q^*\), etcetera. The same argument is valid with respect to RA. Exacly because of this nestedness of the rotated solution, it can be obtained by first computing the first dimension, after which the second dimension is computed (under some orthogonality restriction with regard to the first

\(^2\)Since the CDA solution can be derived from the RA solution after rotation, we only mention PCA and RA in this paragraph.
second dimension is computed (under some orthogonality restriction with regard to the first dimension), after which (if necessary) the third dimension is computed (under some orthogonality restriction with regard to the preceding dimensions), etcetera. As well in PCA as in RA the rotated solution can be found by performing either successive linear PCA’s or successive linear RA’s. In the following paragraph of this paper we describe the PCA approach, which has also been implemented in MULTIPALS.

5.2.1. Rotation of the PCA solution.
For \( t = 1 \) the loss function in (51) can be written in the format of a one dimensional linear PCA

\[
\sigma_1^2(x_1, c_1) = \sum_j \left( q^* j \cdot x_1 c_{j1} \right)'V(j)\left( q^* j \cdot x_1 c_{j1} \right) \tag{53}
\]

where
- \( x_1 \) is the first column of \( X \),
- \( c_1 \) is the first column of \( C \),
- \( c_{j1} \) is the \( j \) th element of \( c_1 \).

After finding a solution for \( x_1 \) and \( c_1 \), by minimizing (53), the second dimension can be found by computing \( Q^{\text{updat}} = Q^* - x_1 c_1' \) and minimizing

\[
\sigma_2^2(x_2, c_2) = \sum_j \left( q^{\text{updat}} j \cdot x_2 c_{j2} \right)'V(j)\left( q^{\text{updat}} j \cdot x_2 c_{j2} \right) \tag{54}
\]

under the restrictions \( u' x_2 = 0 \), \( x_2' x_2 = n \) and \( x_1' x_2 = 0 \). Further dimensions can be found by updating the residuals in \( Q^{\text{updat}} \) and minimizing an analogue of (54).

It should be noted that in the rotational procedure we can replace \( Q^* \) by the matrix \( X^*C'^* \) in which case we obtain the same solution for \( X \) and for \( C \).

5.2.2 Rotation of the RA solution.
Since, under the first normalization restiction, the weighted sums \( X^*A \) are the principal components of \( X^*A^*C'^* \) (that part of \( Q^* \) that is predicted by \( X^* \)) (Fortier, 1966), successive one dimensional linear PCA’s can be applied to obtain a rotated RA solution.

For \( t = 1 \) the loss function in (52) can be written in the format of a one dimensional linear PCA
\[ \sigma_1^2(x^\wedge_1, c_1) = \sum_j \left( X^*A^*c^*_j - x^\wedge_1 c_j \right)V_{ij} \left( X^*A^*c^*_j - x^\wedge_1 c_j \right) \]  

(55)

where
- \( x^\wedge_1 \) is the first principal component of \( X^*A^*C^* \),
- \( c_1 \) is the first column of \( C \),
- \( c_j \) is the \( j \) th element of \( c_1 \).

However, the unrotated solution was computed under the second normalization condition. Thus it seems logical to compute the rotated solution under the second normalization condition also. In that case we do not have the restrictions that \( x^\wedge_l x^\wedge_t = n \), and \( x^\wedge_l x^\wedge_s = 0 \) for \( s < t \), which are usual in PCA. We therefore switch to the equivalent problem of minimizing

\[ \sigma_1^2(x^\wedge_1, c_1) = \sum_i \left( C^*A^*x^*_i - c_1 x^\wedge_i \right)U_{ij} \left( C^*A^*x^*_i - c_1 x^\wedge_i \right) \]  

(56)

where
- \( x^*_i \) is the \( i \) th column of \( X^* \),
- \( c_1 \) is the first principal component of \( C^*A^*X^* \),
- \( x^\wedge_i \) is the first column of \( X^\wedge \),
- \( x^\wedge_i \) is the \( i \) th element of \( x^\wedge_1 \).

Notice that the columns of \( C^*A^*X^* \) and therefore the principal components of (10), \( c_0 \), do not necessarily have zero mean.

After solving (56) for \( x^\wedge_1 \) and \( c_1 \), we proceed by subtracting \( c_1 x^\wedge_1 \) from \( C^*A^*X^* \) and by computing the next principal component as the first principal component of the residuals. Of course, this principal component, \( c_2 \), should be orthogonal to the first principal component, \( c_1 \). After finishing computation for all principal components the matrix \( C \) is directly available. Matrix \( A \) can be computed from

\[ A^*C^* = AC' \]  

(57a)

which implies

\[ A = A^*C^*C \]  

(57b)

After rotation to principal components we can switch to the first normalization by applying a Gram-Schmidt decomposition to \( XA \). Let denote the matrix \( A \) derived from the Gram-Schmidt decomposition of \( XA \) as \( A^{GS} \). Then under the first normalization \( C^{GS} \) is found as
Unlike the rotation of a PCA solution to its principal components, we can not compute the rotated solution of a RA by applying PCA upon $Q^*$. 

5.2.3. Problems in the computation of rotated solutions. 
Due to computational precision the rotated solutions in PCA and RA suffer from an increase in loss with respect to the unrotated solutions. This implies that we do not have a rotation of the principal components. This problem can be solved by projecting each update of a principal component upon the space spanned by the unrotated principal components in the inner majorization loop in which the orthogonality and normalization restrictions are also applied. Moreover, in PCA we recompute $C$ from 

$$C = C^*X^*X$$

This will lead to a perfect rotation of the unrotated solution.

6. Canonical discriminant analysis in MULTIPALS 

6.1. The case of uniform weights. 
Suppose that all elements in $W$, the matrix of data weights, are equal. Then the loss function to be minimized in the redundancy analysis branch of MULTIPALS is:

$$\sigma = \text{tr} \left( (Q - XAB') (Q - XAB') \right)$$  

(59)

where $Q$ is the group indicator matrix of order $n \times k$, with $k$ the number of groups. Also make sure that $Q$ is orthonormalized, i.e. $Q'Q = I$, i.e. all ones are divided by the square root of the number of objects in each group. Furthermore, normalize the reduced rank solution so that $B'B = I$. Then

$$\sigma = k + \text{tr} \left( B'A'XXA \right) - 2 \text{tr} \left( B'Q'X \right)$$

$$= k + \text{tr} \left( A'XXA \right) - 2 \text{tr} \left( B'Q'X \right)$$

$$= k - r + \text{tr} \left( QB - XA \right)' \left( QB - XA \right)$$

(60a)

(60b)

(60c)
and the last term of (60c) is the loss function of the CANALS approach (Gifi, 1981; Van der Burg, 1988) to the problem.

So given this normalizations MULTIPALS will initially compute the same things as REDUNDALS/CANALS. However, for a genuine canonical discriminant analysis solution the matrices A and B should be drastically renormalized.

According to Gifi (1981) discriminant analysis can be approached through solving the auxiliary problem:

\[
\begin{align*}
\text{min } \sigma &= \text{tr } (XA - GB)(XA - GB) \\
A &= B'G'GB = I
\end{align*}
\]  
(61a)

\[
\text{where } G \text{ is the group indicator matrix.}
\]

Setting the partial derivatives with respect to A and B, respectively, equal to zero we obtain:

\[
\begin{align*}
(X'X)A &= X'GB \\
G'XA &= (G'G)B\Lambda
\end{align*}
\]  
(62) \hspace{1cm} (63)

where \( \Lambda \) is a symmetric matrix of Lagrange multipliers arising from the restriction \( B'G'GB = I \).

Suppose \( C \) is defined as \( C = (G'G)^{1/2}B \). Then \( B'G'GB = C'C = I \) and (63) becomes equivalent to:

\[
(G'G)^{-1/2}G'XA = CA.
\]  
(64)

By symmetry of \( \Lambda \) we must have:

\[
C'(G'G)^{-1/2}G'XA = \Lambda = \Lambda' = A'X'G(G'G)^{-1/2}C.
\]  
(65)

This implies that \( C \) must be an orthonormal matrix that symmetrizes \( (G'G)^{-1/2}G'XA \). For solving (63), suppose

\[
(G'G)^{-1/2}G'XA = K\Phi L'
\]  
(66)
Then

\[ C = KL' \]  \hspace{1cm} (67)

and thus, recomputing \( B \) from \( C \), we have:

\[ B = (G'G)^{-1/2}KL' \]  \hspace{1cm} (68)

which satisfies \( B'G'GB = LK'(G'G)^{-1/2}G'(G'G)^{-1/2}KL' = I \) and (63) with

\[ \Lambda = L\Phi L'. \]  \hspace{1cm} (69)

Using (62), (63) and (69) we have

\[ A'X'XA = A'X'GB = \Lambda = L\Phi L' \]  \hspace{1cm} (70)

and therefore

\[ \sigma = k - \text{tr} \Phi \]  \hspace{1cm} (71)

Now define

\[ A^* = AL\Phi^{-1/2} \]  \hspace{1cm} (72)
\[ B^* = BL\Phi^{1/2} \]  \hspace{1cm} (73)

then

\[ A^*X'XA^* = \Phi^{-1/2}L'AX'XAL\Phi^{-1/2} = I \]  \hspace{1cm} (74a)
\[ B^*G'GB^* = \Phi^{1/2}L'B'G'GBL\Phi^{1/2} = \Phi \]  \hspace{1cm} (74b)
\[ A^*X'GB^* = \Phi^{-1/2}L'AX'GBL\Phi^{1/2} = \Phi \]  \hspace{1cm} (74c)

So

\[ \sigma (A^*, B^*) = k - \text{tr} \Phi. \]  \hspace{1cm} (75)

Notice that this is the normalization of the first normalization condition of the redundancy analysis branch of MULTIPALS. This is a normalization on the total sums of squares and
products, whereas we would prefer normalization on the pooled within sums of squares. Eliminating B from the stationary equations (62) and (63) yields:

$$X'G(G'G)^{-1}G'XA = X'XA\Lambda$$  \hspace{1cm} (76)

and this implies for the "total-normalized" A*:

$$X'G(G'G)^{-1}G'XA^* = X'XA^*\Phi.$$  \hspace{1cm} (77)

Now define

$$\Psi = \Phi (I - \Phi)^{-1}$$  \hspace{1cm} (78)

and

$$K \triangleq A^* (I + \Psi)^{1/2} = A^* (I - \Phi)^{-1/2}.$$  \hspace{1cm} (79a)

Then it may be verified that

$$K'X'K = (I - \Phi)^{-1/2}A^*X'X(A^* - \Phi)^{-1/2} = (I - \Phi)^{-1}$$  \hspace{1cm} (80a)

$$K'X'G(G'G)^{-1}G'XK = \Phi (I - \Phi)^{-1}.$$  \hspace{1cm} (80b)

Thus for K we get the "within normalization":

$$K'WK = I$$  \hspace{1cm} (81)

and from (77) and (79a) it follows that:

$$X'G(G'G)^{-1}G'XK(I + \Psi)^{-1/2} = X'XK(I + \Psi)^{-1/2}\Psi(I + \Psi)^{-1}$$  \hspace{1cm} (82)

so that we have:

$$X'G(G'G)^{-1}G'XK = [X'X - X'G(G'G)^{-1}G'X]K\Psi.$$  \hspace{1cm} (83)

Hence
\[ BK = WK'P, \quad (84) \]

the stationary equations of canonical variate analysis, which is the \( k > 2 \) group generalization of Fisher’s (1936) discriminant analysis technique, usually derived from maximizing the ratio of Between- to Within-group variance. In particular, setting

\[ \Psi(K) = \frac{K'BK}{K'WK} \quad (85) \]

the partial derivatives of \( \Psi'(K) \) are:

\[ \frac{\delta \Psi(K)}{\delta K} = (K'WK)(2BK) - (K'BK)(2WK). \quad (86) \]

Thus (84) arises from repeatedly setting (86) equal to zero. Canonical variate analysis is a PCA on the group means after the pooled within dispersion has been made spherical. It is of some interest to note that in terms of the loss function the canonical variate analysis solution satisfies:

\[ \sigma(K) = \text{tr} \; K'XXK - \text{tr} \; K'X'(G'G)^{-1}G'XK \]
\[ = \text{tr} \; (I - \Phi)^{-1} - \text{tr} \; \Phi(I - \Phi)^{-1} = k. \quad (87) \]

6.2. The case of non-uniform weights.

Suppose that the matrix of data weights, \( W \), has identical columns. Then the redundancy analysis branch of MULTIPALS can be written as follows:

\[ \sigma_W = \text{tr} \; (Q - XA')'V(Q - XA') \quad (88) \]

where \( V \) is an \( n \times n \) diagonal matrix with the elements of a column of \( W \) on the diagonal.

Now minimization of (88) will also minimize:

\[ \text{tr} \; (QB - XA)'V(QB - XA) \quad (89) \]

if

\[ \text{tr} \; Q'VQ - \text{tr} \; B'Q'VB = c \quad (90) \]
with $c$ some constant, for all possible matrices $Q, B, X, A$ and $V$. This will be true if we require $Q'VQ = I$ and $B'B = I$. Consequently, a renormalization of $A$ and $B$ should be computed such that the weighted pooled within dispersion becomes spherical.

In addition, it seems interesting to investigate whether it is possible to solve an extension of canonical variate analysis, in which $W$ is allowed to have different columns, by using MULTIPALS. This extension might for instance be useful in cases where we do not know to which group an object belongs, though we have information that the object does not belong to some of the $k$ groups.
References


Proposed syntax for MULTIPALS

Overview of commands

TITLE title
SUBTITLE subtitle
DATASET filename
VARIABLES varlist
FORMAT (format)
GROUPS groupname1 = varlist1 [/ groupname2 = varlist2 / ...]
ANALYSIS REGR / PATH / PCA = varlist / ALL
DEPENDENT varlist
ACTIVE varlist / ALL
PASSIVE varlist
LEVEL varlist / ALL (NOMI / ORDI / NUME / POLYx)
SPLINES varlist / ALL (TYP1 / TYP2 / TYP3, knots)
CATEGORIES varlist / ALL (FIXED / EQUAL, n / NORMAL, n / CONTINU)
MISSING varlist / All (OMIT / SCAT / MCAT, value)
DIMENSIONS highest, lowest
SOLUTION INITIAL / FINAL / BOTH
ITERATE n
TOLERANCE n
PRINT FREQ / HIST / EIGEN / WEIGHTS / OBJECT / QUANT / CORREL / NO
PLOT HIST / WEIGHTS / OBJECT [(varlist)] / QUANT [(varlist)] / TRANSF [(varlist)] / NO
WRITE WEIGHTS / OBJECT / QUANT / TRDATA / NO
SPECIALS 1 / 2 / 3 / 4 / 5

Set-up

The commands DATASET, VARIABLES, FORMAT and ANALYSIS are obligatory. The command DEPENDENT is only obligatory when regression- or path-analysis is specified on the analysis-command. Every non-obligatory command has a default. The command ANALYSIS must appear before the commands DEPENDENT and ACTIVE (if used) and all three of these commands should appear before the commands LEVEL and DIMENSIONS. This is necessary because the first three commands invoke several defaults which override choices made with the commands LEVEL and DIMENSIONS.

Each keyword may be abbreviated to four characters. The keyword VARIABLES may be abbreviated to either VARI or VARS. The keyword that specifies the requested type of analysis (regression, path or principal components) may be abbreviated to three characters (REG, PAT or PCA). So may the keyword that specifies whether results should be outputted for the initial, the final or both solutions. The keywords on the PRINT-, PLOT- and WRITE-commands may even be abbreviated to two characters.

The program is invoked on the VAX by typing RUN MULTIPALS. The program then asks for the file that contains the set-up commands. The first nine characters of this filename or the characters up to a dot in the filename (whichever comes first) are used for the name of the file that contains the output of the job. The extension of this output-file is '.LST'.

Description of commands

TITLE title

Description or name of the research-subject. A maximum of 75 characters is allowed for this description.
Default = no title.
subtitle
Name of the analysis, max. 75 characters.
Default = no subtitle.

filename
Name of the file that contains the data to be analyzed. The maximum length is 13 characters.

varlist
List of names of variables. Names may be no longer than 8 characters. The keyword 'TO' may be used to imply variables: VAR001 TO VAR006 implies six variables. If 'TO' is used, the first character of the names of the variables on both sides of this keyword must be the same letter and the last one a number.

(format)
Contains a permissible Fortran-format within parentheses.

groupname1 = varlist1 [ [ / groupname2 = varlist2 ] ... ]
Variables may be arranged in groups for easier reference in later commands. The keyword 'TO' on this and following commands refers to the complete list of variables that is specified above and is not restricted to variables beginning with the same letter(s). Every list of variables must be in the same sequence as the list on the VARIABLES-command.
Default = no groups specified.

REGR / PATH / PCA = varlist / ALL
The first part of this command specifies the type of analysis to be performed; the second part the variables to be involved in the analysis. Groups may be part of this variable list, but may not be part of TO-lists.

varlist
One or more dependent variables must be specified when a regression- or path-analysis is requested.

varlist / ALL
List of variables to be analyzed. ALLLL refers to the variables specified on the ANALYSIS-command.
Defaults = REG and PAT: ALL minus DEPENDENT; PCA: ALL

varlist
List of variables that are not part of the analysis, but which can be used to label plots of objectscores and for which category-quantifications will be computed if requested with the SPECIALS-command.
Default = no passive variables.

varlist / ALL (NOMI / ORDI / NUME / POLYx)
Measurement level of all the variables in the analysis.
NOMI = nominal
ORDI = ordinal
NUME = numerical = linear
POLYx = polynomial, the value of x specifies the degree of the polynomial and may be no higher than 4.
Default = ALL (ORDI)

varlist / ALL (TYP1 / TYP2 / TYP3, knots)
For the moment the names of the types of splines are set to TYP1, TYP2 and TYP3.
Default = no splines.
CATEGORIES

varlist / ALL (FIXED / EQUAL, n / NORMAL, n / CONTINU)

FIXED = every value of the variable is a separate category.
EQUAL = the variable is to be recoded such that n new categories with equal (or as equal as possible) numbers of objects in each category arise.
NORMAL = the variable is to be recoded such that n new categories arise which form the best possible approximation of a normal distribution.
CONTINU = the categories form a continuous scale.
Default = ALL (FIXED)

MISSING

varlist / ALL (OMIT / SCAT / MCAT, value)

OMIT = objects with a missing value are omitted from computations involving that variable.
SCAT = missing values are treated as a separate category.
MCAT = multiple categories: every occurrence of a missing value is treated as a new category.
Default = no missing values specified

DIMENSIONS

highest, lowest
The number of dimensions solutions are to be computed for. If both numbers are the same, only one solution is computed.
Default = REG and PAT: (2,1); PCA: (3,1)

SOLUTION

INITIAL / FINAL / BOTH
Specifies whether results from the initial, the final or both solutions should be printed. The next five commands can be separately specified for the initial and for the final solutions.
Default = FINAL

ITERATE

maximum number of iterations
Defaults = INITIAL: 50; FINAL: 75

TOLERANCE

minimal improvement of fit
This number must be between 0 and 1, but may not be smaller than 0.00000001 (1E-08 (?)).
Defaults = INITIAL: 0.05/#cases; FINAL: 0.00001

PRINT

FREQ / HIST / EIGEN / WEIGHTS / OBJECT / QUANT / CORREL / NO
FREQ = frequencies of categories of variables
HIST = history of iterations
EIGEN = eigenvalues
WEIGHTS = REG and PAT: regressionweights; PCA: componentloadings
OBJECT = objectscores
QUANT = category quantifications
CORREL = matrix of correlations between variables
NO = no results are printed
Defaults = INITIAL: NO; FINAL: EIGEN QUANT

PLOT

HIST / WEIGHTS / OBJECT (varlist) / QUANT (varlist) / TRANSF (varlist) / NO
HIST = history of iterations.
WEIGHTS = weights or loadings (cf. PRINT).
OBJECT = objectscores; when no variablenames are present, an unlabeled plot is made; otherwise plots of objectscores labeled with the requested variables are made.
QUANT = category quantifications; when no variablenames are present, all quantifications are plotted in the
same picture; otherwise separate plots of different variables are made.

TRANSF = new category quantifications plotted against original category numbers; when no variable names are present, all transformations are plotted in the same picture; otherwise separate plots of different variables are made.

NO = no results are plotted.
Defaults = INITIAL: NO; FINAL: OBJECT QUANT

WRITE

WEIGHTS / OBJECT / QUANT / TRDATA / NO
WEIGHTS = weights or loadings are written to 'file'.LDN
OBJECT = objectscores are written to 'file'.SCO
QUANT = category quantifications are written to 'file'.QUA
TRDATA = the transformed datamatrix, in which the original category numbers are replaced by the new category quantifications, is written to 'file'.TRD

NO = no results are written to files.
'File' is the same as the first part of the name of the file that contains the output of the job. If both results from the initial and from the final configuration are requested, the results from the final configuration are appended to the results from the initial configuration.
Defaults: NO

SPECIALS

1 / 2 / 3 / 4 / 5
1 = overview of specified job
2 = comments and hints in output-file
3 = computation of (final) category quantifications of passive variables
4 = crossvalidation on passive objects
5 = orthonormalization (?)
Defaults: no specials.

What is missing

The above commands do not yet provide the possibility to specify whether the analysis has to be row- or column-conditional. Neither is it possible to designate objects as passive.

For some applications it might be useful to be able to start computations from a given configuration. That is not yet incorporated either.

Plots are made for every pair of dimensions in the solution. The user should be able to request specific pairs of dimensions.