PART II

INDOQUAL AND INDOMIX

5. INDORT FOR QUALITATIVE VARIABLES (INDOQUAL)

5.1. Introduction

In chapter 4 a review has been given of methods for the analysis of qualitative and quantitative variables, based on applying three-way methods to a set of quantification matrices for the variables. This idea has first been worked out for the particular case of the analysis of qualitative variables by Kiers (1989c). He considered the application of INDORT on quantification matrices $S_j = (m_j-1)^{-1/2}JG_jD_j^{-1}G_j'J$, $j = 1,\ldots,m$. In the present chapter a different choice of quantification matrices is made. That is, the application of INDORT to non-normalized quantification matrices $S_j = JG_jD_j^{-1}G_j'J$ will be discussed. In the sequel this method is called "INDOQUAL" (**IND**scal with **O**rthonormality constraints applied to quantification matrices for **QUAL**itative variables). In the present chapter parts of the results given by Kiers (1989c) are repeated (adapted to the different choice of a quantification matrix), and some additional results on this method are considered.

As has been mentioned in chapter 1, two different types of methods are available for PCA of qualitative variables. The first type of method is PCA of quantification matrices, which comes down to performing PCA of the matrix of correlations between the variables. This method aims at optimally representing the variables, whereas it does not give coordinates for the objects at all. Cazes, Bonnefous, Baumerder and Pagès (1976) extended the method for PCA of qualitative variables proposed by Saporta (1975). Their method does provide object coordinates. However, the object coordinates found in their method are based only on the first principal component of the variables. Hence it can be concluded that PCA of quantification matrices does not adequately find coordinates for the objects.

The second type of method for PCA of qualitative variables is mainly represented by Multiple Correspondence Analysis (MCA). This method can be seen as PCA of the categories and objects, but does not necessarily represent the variables well. If one's main interest is in the representation of qualitative variables and one does not need any information on how the relations between variables are reflected in relations between objects and categories one might be satisfied with the representation of the variables by the first type of method, PCA of quantification matrices. On the other hand, MCA should be used if one is mainly interested in representing the categories and the objects. Clearly, both methods are incomplete in that they lack either an optimal representation of the objects (PCA of quantification matrices), or an optimal representation of the association between the variables (MCA). In section 4.1 it has been shown that INDOQUAL can be seen as a compromise between PCA of quantification matrices (using ϕ^2–coefficients) and MCA. In the present chapter it will be shown in a different way that INDOQUAL can be seen as such a compromise, and some of its implications will be discussed. In addition, some properties that are well–known for MCA are shown to hold for INDOQUAL too. In a final section it is shown that INDOQUAL has some relations with a method proposed by Saporta (1979). First, however, INDOQUAL will be described.

5.2. INDORT for qualitative variables (INDOQUAL)

In chapter 2 the three–way method INDORT has been described as the method that finds matrices of object coordinates X $(n \times r)$ and diagonal matrices W_j $(r \times r)$ such that the loss function

$$\sigma(X,W_1,\ldots,W_m) = \sum_{j=1}^{m} \| S_j - XW_jX' \|^2 \qquad (1)$$

is minimized over X and $W_1,\ldots,W_m$, subject to $X'X = I_r$. The problem of minimizing (1) over X and $W_1,\ldots,W_m$, subject to $X'X = I_r$, can be simplified as follows. In order to minimize σ over W_j for fixed X, one only has to minimize $\|S_j - XW_jX'\|^2$ over W_j, $j = 1,\ldots,m$. This term can be expanded as

$$\| S_j - XW_jX' \|^2 = \text{tr } S_j^2 + \| X'S_jX - W_j \|^2 - \text{tr } X'S_jXX'S_jX. \qquad (2)$$

The right hand–side of (2) only contains W_j in its second term, which, is minimized by choosing W_j as $\text{Diag}(X'S_jX)$.

Having found the solution for minimizing σ over $W_1,\ldots,W_m$ in terms of X, one can express σ as a function to be minimized over X only, by substituting $W_j = \text{Diag}(X'S_jX)$ for every j. This results in

$$\sigma(X) = \sum_{j=1}^{m} \| S_j - X\,\mathrm{Diag}(X'S_jX)\,X' \|^2$$

$$= \sum_{j=1}^{m} \mathrm{tr}\; S_j^2 \;-\; \sum_{j=1}^{m} \mathrm{tr}\; S_jX\,\mathrm{Diag}(X'S_jX)\,X'. \tag{3}$$

Obviously, minimizing $\sigma(X)$ over X subject to $X'X = I_r$ is equivalent to maximizing

$$\mathrm{f}(X) = \sum_{j=1}^{m} \mathrm{tr}\; X'S_jX\;\mathrm{Diag}(X'S_jX) \tag{4}$$

over X subject to $X'X = I_r$. It is useful to note that $\mathrm{f}(X)$ can equivalently be expressed as

$$\mathrm{f}(X) = \sum_{j=1}^{m} \mathrm{tr}\left[\mathrm{Diag}(X'S_jX)\right]^2$$

$$= \sum_{j=1}^{m} \sum_{l=1}^{r} (\mathbf{x}_l'S_j\mathbf{x}_l)^2, \tag{5}$$

where $\mathbf{x}_l$ denotes the l^{th} column of X.

Kiers (1989c) uses the algorithm proposed by Ten Berge, Knol, and Kiers (1988) for maximizing $\mathrm{f}(X)$, subject to $X'X = I_r$. This algorithm becomes problematic when n is large. In chapter 9 a modification of this algorithm is described which does not depend on the size of n.

Kiers (1989c) proposed to use the normalized quantification matrix $S_j = (m_j-1)^{-1/2}JG_jD_j^{-1}G_j'J$. This has been done in order to facilitate comparing INDORT with PCA of Tschuprow's T^2–coefficients (Saporta, 1975). However, using the non–normalized quantification matrices,

$$P_j = JG_jD_j^{-1}G_j'J \tag{6}$$

does not jeopardize the possibility of comparing INDORT and PCA of quantification matrices. This can be seen as follows. In Table 4.1, PCA of Tschuprow's T^2–coefficients and the method proposed by Kiers (1989c) are in the same row, being applications of three–way methods to the same quantification matrices. Similarly, PCA of ϕ^2–coefficients (Escoufier, 1980) and INDOQUAL are in the same row, implying that hierarchical relations between these methods exist, just as those between PCA of Tschuprow's T^2–coefficients and the method proposed by Kiers (1989c). In addition, comparing INDOQUAL and MCA is, in fact, easier than comparing MCA and the method proposed by Kiers (1989c), because MCA and INDOQUAL are in the same row of Table 4.1.

In section 4.4 some remarks have been made about whether or not to use normalized quantification matrices. In the specific case of INDORT applied to quantification matrices for qualitative variables, using normalized quantification matrices may yield a solution that is dominated, to a certain extent, by variables with small numbers of categories, particularly, by dichotomous variables. An example and an explanation for this are given in section 10.6. It should be noted, incidentally, that normalizing the quantification matrices has no effect whatsoever on the INDORT solution when the variables have equal numbers of categories.

5.3. INDOQUAL as a compromise between MCA and PCA of φ^2-coefficients

INDOQUAL can be interpreted in a number of different ways. It will be shown here that INDOQUAL can be interpreted as a method that optimally represents the variables (as PCA of quantification matrices does) while retaining a clear link with the representation of the objects. First, it is useful to describe PCA of ϕ^2–coefficients (Escoufier, 1980) in mathematical terms.

As has been remarked in section 4.1, PCA of ϕ^2–coefficients can be seen as PCA of the quantification matrices $P_j \equiv JG_jD_j^{-1}G_j'J$ considered as variables. That is, the matrices P_j can be strung out row–wise into column–vectors Vec(P_j), and PCA can be performed on the resulting data matrix. It is well–known that PCA maximizes the sum of squares of loadings of the variables on the components. In general, a loading is a signed length of the projection of a variable on a component, expressing the amount of inertia of the variable accounted for by the component. Usually, the loadings in PCA are product–moment correlations. PCA of ϕ^2–coefficients can be seen as a kind of "raw" PCA, that is, it is a PCA of non–normalized variables. In this case the "loading" of variable j on component l is given by tr $F_l'P_j$, where the $(n \times n)$ matrix F_l is obtained from Vec(F_l), the l^{th} principal component of the vectors Vec(P_1),...,Vec(P_m). Hence PCA of ϕ^2–coefficients can be described as maximizing the function

$$g(F_1,...,F_r) = \sum_{j=1}^{m} \sum_{l=1}^{r} (\text{tr } F_l'P_j)^2, \tag{7}$$

over the $n \times n$ matrices F_l, $l = 1,...,r$, subject to the constraint $\text{Vec}(F_l)'\text{Vec}(F_{l'}) = \text{tr } F_l'F_{l'} = \delta_{ll'}$, where δ denotes the Kronecker symbol.

The function maximized by INDOQUAL is closely related to the function maximized by PCA of ϕ^2–coefficients, as can be seen as follows. INDOQUAL has been shown to maximize (5), which upon substitution of P_j for S_j can be rewritten as

$$f(X) = \sum_{j=1}^{m} \sum_{l=1}^{r} (\mathbf{x}_l'P_j\mathbf{x}_l)^2 = \sum_{j=1}^{m} \sum_{l=1}^{r} (\text{tr } \mathbf{x}_l\mathbf{x}_l'P_j)^2, \tag{8}$$

over matrix X, subject to $X'X = I_r$. Maximizing $g(F_1,...,F_r)$ over $F_1,...,F_r$, subject to the constraint $\text{tr } F_l'F_{l'} = \delta_{ll'}$ and subject to the additional constraint $F_l = \mathbf{x}_l\mathbf{x}_l'$, is equivalent to maximizing $f(X)$ over X, subject to the constraint $\text{tr}(\mathbf{x}_l\mathbf{x}_l'\mathbf{x}_{l'}\mathbf{x}_{l'}') = \delta_{ll'}$. The latter constraint can be reformulated as $\text{tr}(\mathbf{x}_l\mathbf{x}_l'\mathbf{x}_{l'}\mathbf{x}_{l'}') = (\mathbf{x}_l'\mathbf{x}_{l'})^2 = \delta_{ll'}$. This in turn is equivalent to $X'X = I_r$, which shows that, when $F_l = \mathbf{x}_l\mathbf{x}_l'$, the constraints $\text{tr } F_l'F_{l'} = \delta_{ll'}$ and $X'X = I_r$ are equivalent. As a consequence, maximizing $g(F_1,...,F_r)$ over $F_1,...,F_r$, subject to the constraint $\text{tr } F_l'F_{l'} = \delta_{ll'}$, for all pairs l and l', and to the additional constraint $F_l = \mathbf{x}_l\mathbf{x}_l'$ for all l, is equivalent to maximizing (8) over X, subject to $X'X = I_r$. Hence INDOQUAL can be interpreted as PCA of ϕ^2–coefficients subject to the additional constraint $F_l = \mathbf{x}_l\mathbf{x}_l'$.

An advantage of INDOQUAL over PCA of ϕ^2–coefficients is that, whereas in the latter the components that are found cannot immediately be interpreted in terms of the objects, this problem is overcome in INDOQUAL. This is due to the constraint imposed on the "components", $F_1,...,F_r$, for the qualitative variables. The constraint $F_l = \mathbf{x}_l\mathbf{x}_l'$ implies that for every component of the variables there is one vector $\mathbf{x}_l$ of coordinates for the *objects*. Therefore, in INDOQUAL the components solution is directly linked to a solution for object coordinates on the same number of dimensions as one has chosen for the components of the variables. A representation of the categories is immediately supplied by means of the centroids of the object coordinates of the objects that score in a category.

As far as comparison of INDOQUAL and MCA is concerned, one finds that INDOQUAL yields a representation of the qualitative variables that is better than the one given by MCA, because the MCA solution satisfies the same constraints as are imposed on the components in INDOQUAL, and INDOQUAL yields the best possible representation of the variables, subject to these very constraints. This also follows immediately from the fact that MCA can be seen as SUMPCA on the quantification matrices P_j, and that SUMPCA is a constrained variant of INDORT, as has been explained in chapter 2.

In conclusion, INDOQUAL can be seen as a method that optimally represents relations between the qualitative variables, and simultaneously yields a representation of the objects and the categories that is linked to the representation of the variables. Clearly, INDOQUAL is a compromise between PCA of ϕ^2-coefficients and MCA, in that it performs a PCA of the variables (but subject to additional constraints) on the one hand, and yields coordinates for the objects (like MCA does) on the other hand.

Although INDOQUAL and MCA clearly differ in terms of the objectives they have, the methods do have certain properties in common. One of these is discussed in the next section.

5.4. Trivial solutions

A well-known property of MCA is that, when MCA is seen as finding the object coordinates as the eigenvectors of $\sum_j S_j = \sum_j G_j D_j^{-1} G_j'$, one always finds one so-called trivial axis of object coordinates (e.g., Gifi, 1981, p.94). That is, $\sum_j G_j D_j^{-1} G_j'$ always has an eigenvector 1, associated with the largest eigenvalue, regardless of the data. In order to avoid such a solution, one may obtain the object coordinates from the first r eigenvectors of $\sum_j J G_j D_j^{-1} G_j' J$, which are precisely the first r non-trivial eigenvectors of $\sum_j G_j D_j^{-1} G_j'$.

In INDOQUAL the same phenomenon is observed. That is, INDORT applied to the matrices $G_j D_j^{-1} G_j'$ yields one trivial axis, and the same (non-trivial) axes as those of INDORT applied to $J G_j D_j^{-1} G_j' J$. This phenomenon has occurred consistently in practice, but a mathematical proof is not available. It can be proven, however, that, if the trivial vector emerges, the remainder of the r+1 dimensional solution gives an r-dimensional solution for the centered case.

5.5. The interpretation of the results of an INDOQUAL analysis

The results of an INDOQUAL analysis can be interpreted in a way highly similar to that of MCA. The difference in interpretation should be that the different methods stress different aspects. That is, whereas MCA stresses optimal representation of the categories, INDOQUAL stresses optimal representation of the variables.

First of all, we have the diagonal elements of the matrices W_j. These can be interpreted as the loadings of the qualitative variables on the components. Note that these loadings are always nonnegative, due to the fact that $W_j = \text{Diag}\, X'S_jX$, and S_j is positive semi-definite. This might seem to restrict the quality of these loadings, but, considering that relations between qualitative variables cannot sensibly be expressed in terms of negative correlations, the nonnegativity of these loadings merely reflects the inappropriateness of negative correlations for qualitative variables (Janson & Vegelius, 1982).

The loading of variable j on component l is given by $\mathbf{x}_l'S_j\mathbf{x}_l$. This measure is identical to what is called "discrimination measure" by Gifi (1981). It is readily verified that this is in fact the η^2-coefficient between the qualitative variable j and the (quantitative) component l. The maximal value of η^2 is unity. The total inertia of a qualitative variable is given by $\text{tr}\, S_j^2 = (m_j-1)$, hence, when variable j has more than two categories, the total inertia of this variable is larger than 1, and it can never be accounted for completely by means of one component. This might seem to be a disadvantage, but in practice this poses no problems. The fact that a variable can never be accounted for completely by a component can be understood by noting that a variable may in fact incorporate several different relations between categories, which cannot be captured in one dimension.

For the interpretation of a component it is useful to note that the "loading" is one if and only if the component perfectly discriminates the objects in terms of the categories to which they belong. That is, it is one if and only if objects that belong to the same categories have the same scores on that component. Therefore, high loadings of certain variables on a component imply that the component discriminates the objects well in terms of the categories of the variables concerned. In order to see which categories are particularly well discriminated, it is useful to compute the mean scores of the components in each of the categories of the variables. These are given by the elements of $Y_j = D_j^{-1}G_j'X$, $j = 1,...,m$.

In addition to the W_j, we have the object coordinates matrix X. These coordinates represent the objects in a low-dimensional space, taking into account the main relations between the qualitative variables.

Finally, we have an overall value for evaluating the quality of the solution. To this end, we use the proportion of inertia accounted for of the quantification matrices P_j. This proportion is obtained as the maximal value of f(X) (cf.(8)), divided by the total inertia. The total inertia of matrix P_j is equal to (m_j-1), hence the overall total inertia is equal to $(\sum_j m_j - m)$. Therefore, the proportion of inertia accounted for is given by

$$\frac{\sum_{j=1}^{m} \sum_{l=1}^{r} (\mathbf{x}_l{}'P_j\mathbf{x}_l)^2}{(\sum_j m_j - m)} . \tag{9}$$

5.6. A relation of INDOQUAL with a method proposed by Saporta

Above, INDOQUAL has been described as an alternative to MCA. This is by no means the only alternative to MCA that has been considered in the literature. One of the alternatives to MCA greatly resembles INDOQUAL, as will be shown here.

Saporta (1979) has described several techniques for analyzing qualitative variables. The objectives of these techniques are to search weights for the variables such that they yield an optimal (weighted) MCA solution, in one or more dimensions. One of his methods consists of maximizing the first eigenvalue of $\sum_j\alpha_jP_j$, subject to $\sum_j\alpha_j^2 = 1$. This is equivalent to maximizing $\mathbf{x}'\sum_j\alpha_jP_j\mathbf{x}$ over α_j and $\mathbf{x}$, subject to $\sum_j\alpha_j^2 = 1$ and $\mathbf{x}'\mathbf{x} = 1$. As he shows, the α_j, $j = 1,...,m$, that maximize this function are given by $\alpha_j = \mathbf{x}'P_j\mathbf{x}/(\sum_j(\mathbf{x}'P_j\mathbf{x})^2)^{½}$. As a result, his method comes down to maximizing $\sum_j(\mathbf{x}'P_j\mathbf{x})^2/(\sum_j(\mathbf{x}'P_j\mathbf{x})^2)^{½} = (\sum_j(\mathbf{x}'P_j\mathbf{x})^2)^{½}$, subject to $\mathbf{x}'\mathbf{x} = 1$. Clearly, this is equivalent to maximizing $\sum_j(\mathbf{x}'P_j\mathbf{x})^2$, subject to $\mathbf{x}'\mathbf{x} = 1$. Hence this method is equivalent to INDOQUAL, with $r = 1$.

Saporta (1979) has generalized his method to obtain more than one dimension. For the case of more than one dimension he proposed to maximize

$\sum_l\lambda_l(\sum_j\alpha_jP_j) = \sum_l\mathbf{x}_l'(\sum_j\alpha_jP_j)\mathbf{x}_l = \text{tr}\,X'(\sum_j\alpha_jP_j)X$, where $\lambda_l(.)$ denotes the l^{th} eigenvalue of the matrix between brackets. The solution for α_j is now given by $\alpha_j = \text{tr}\,X'P_jX/(\sum_j(\text{tr}\,X'P_jX)^2)^{½}$. Then his method comes down to maximizing $\sum_j(\text{tr}\,X'P_jX)^2/(\sum_j(\text{tr}\,X'P_jX)^2)^{½} = (\sum_j(\text{tr}\,X'P_jX)^2)^{½}$. Clearly, maximizing this function over column-wise orthonormal X for $r > 1$ is not generally equivalent to maximizing the function $\sum_j\sum_l(\mathbf{x}_l'P_j\mathbf{x}_l)^2$, as INDOQUAL does.

The above comparison of INDOQUAL to the Saporta (1979) methods leads to another interpretation of INDOQUAL. As follows from the above, in case $r = 1$, INDOQUAL can be interpreted as the method that finds those weights for the variables that yield the best one-dimensional (weighted) MCA solution. When $r > 1$, a similar interpretation for INDOQUAL can be given. That is, INDOQUAL can be seen as the method that simultaneously finds weights w_{jl} for the variables *and* the components such that each component can be seen as a kind of one-dimensional weighted MCA solution, with the different MCA-solutions constrained such that they are mutually orthogonal. The weights for the variables differ for the different components. When the loadings are close to zero or one, INDOQUAL can be seen as a method that combines several one-dimensional MCA solutions on several subsets of variables. These subsets are defined by the variables that load high on a component. Therefore, INDOQUAL may be a particularly useful alternative to MCA for data with two or more subsets of closely related variables.

5.7. Discussion

In the present chapter, INDOQUAL has been described as a method for PCA of qualitative variables which does not only yield loadings for the variables, but also object coordinates that are linked to the loadings of the variables. It has been shown that MCA is less suitable for such a purpose than INDOQUAL. However, as has been discussed in section 4.4, the choice of a method should not only be based on its adequacy in representing the information in one's data. Another measure for the quality of a method is parsimony of the model. As described in section 2.6, there is a trade-off between parsimony of the model and fit of the model. In general, a useful strategy seems to choose the simplest model that provides a sufficient fit for the quantification matrices.

Apart from the above considerations in choosing between MCA and INDOQUAL, one might see the fact that INDORT finds "unique axes" as a particular advantage of INDOQUAL over MCA. That is, the INDOQUAL solution is determined completely, whereas the MCA solution is determined up to a rotation only. In chapter 8, however, a procedure for rotating the MCA solution so as to maximize similar criteria as maximized by INDOQUAL is discussed. In this way, "unique axes" can be determined for the MCA solution as well.

In section 5.1 it has been remarked that the object coordinates provided by the Cazes et al. method are not very useful, because they are based only on the first principal component of the variables. It is possible, however, to adapt the Cazes et al. method such that it provides object coordinates dimensions for each of the principal components. In a different context D'Alessio (1988) describes such a procedure. He also remarks that this method treats the different aspects of the data asymmetrically. In the present context this implies that the representation of the variables receives most attention, while the representation of the objects is of only secondary importance. In addition, the loadings found by the original PCA are related to the corresponding object coordinates in a less direct way than is the case in INDOQUAL.

6. SOME ADDITIONAL COMPARISONS OF MCA AND INDOQUAL

Above, a variety of methods for the analysis of qualitative variables has been discussed. Among these methods, INDOQUAL has received special attention in chapter 5. As has been mentioned in chapter 5, INDOQUAL uses the same quantification matrices as MCA does (implicitly). In chapter 5 INDOQUAL has been described as a compromise between Escoufier's method for PCA of ϕ^2–coefficients (Escoufier, 1980) and MCA. As far as the interpretation of the methods is concerned, only one difference has been discussed in chapter 5. In the present chapter, some further differences and equivalences in terms of the objectives of the methods will be discussed. That is, a number of different descriptions of the methods will be given, showing that the methods differ from each other in more than one respect.

6.1. A comparison of MCA and INDOQUAL in terms of PCA of qualitative variables

A well–known interpretation of MCA is the following. Each qualitative variable can be considered as a quantitative variable when all categories of the variable receive certain quantitative values. This process of assigning quantitative values to the categories of qualitative variables results in so–called "quantified variables". For the first MCA component, the variables are quantified such that the first eigenvalue of the matrix of correlations between the quantified variables is maximized over possible quantifications (not to be confused with quantification matrices) of the variables. The object scores corresponding to this solution are proportional to the mean of the quantified variables. MCA finds a second component by searching *other* quantifications for the variables that yield a vector of object scores (again proportional to the mean of the quantified variables) which is orthogonal to the first object scores component, and maximally accounts for the inertia of the corresponding quantified variables. Subsequent components are found in a similar way.

Gifi (1981, p.103–104) has shown that the first MCA component in fact yields the one–dimensional PCA solution for the set of optimally quantified variables. Gifi (1981) has also mentioned that subsequent MCA components do not generally yield the one-dimensional PCA solutions for the sets of variables quantified according to these components. Therefore, it seems that MCA can be interpreted as a method for PCA of the (optimally quantified) variables only as far as the first component is concerned.

The problem of interpreting MCA as a PCA of optimally quantified variables seems to be that in MCA each single variable is represented by a number of different quantitative variables (that are seen as quantified versions of the original qualitative variables). Obviously, this takes into account the diverse information possibly present in a qualitative variable to a certain extent. However, each qualitative variable is represented by the same number (r) of quantitative variables, whereas it is conceivable that for some variables that carry more information more quantitative variables would be necessary, while for other variables representing them by r quantitative variables might be superfluous. This problem is remedied by De Leeuw and Van Rijckevorsel (1988) who propose a variant of MCA that allows for different numbers of quantified variables (called *copies*) to represent different variables. However, for both MCA and the variant of MCA proposed by De Leeuw and Van Rijckevorsel, to each quantification corresponds a different correlation matrix, and hence one cannot see these methods as methods performing just one PCA of quantified variables. In this way it becomes also rather awkward to interpret the so-called "discrimination measures", $\mathbf{x}_l'P_j\mathbf{x}_l$, $j = 1,...,m$, $l = 1,...,r$, as squared loadings of quantified variables on the components as is done in MCA (Gifi, 1981, p. 96–97). As far as only one component is concerned, this interpretation is warranted, but as soon as higher dimensional solutions are considered the "squared component loadings" cannot be compared over different components, because, in fact, different quantitative variables (that is different quantifications of qualitative variables) are involved. Therefore, MCA cannot generally be seen as a method for PCA of (optimally) quantified variables.

A method which is directed more clearly at PCA of qualitative variables (or any type of variables) has been proposed by Young, Takane & De Leeuw (1978), under the name PRINCIPALS, which is also an option of the PRINCALS program (De Leeuw and Van Rijckevorsel, 1980). In this method the qualitative variables are each replaced by one "quantified variable" and this is done in such a way that the resulting PCA solution explains as much variance as possible. Obviously, this method does allow for interpretation in terms of a

PCA of quantified variables, unlike MCA. It has a different problem, however, in that it replaces each qualitative variable by only one quantified variable. In case the qualitative variables at hand can be considered as quantitative variables that have for some reason (like measurement problems) been polytomized, then PRINCIPALS may be a useful method, assuming that it retrieves the correct quantifications for the variables (which is still a matter of doubt, because the only criterion on which the search for quantifications is based pertains to the correlations of a variable with the other variables, thus depending on which other variables are included in the analysis). However, for any variable that contains information which cannot be captured by means of one quantitative variable, the PRINCIPALS approach seems to be inappropriate.

A PCA technique that does take into account all the information present in the variables, and does not loose information by considering only one (or even several) quantification(s) of the qualitative variables is what has been called PCA of quantification matrices. As has been mentioned in chapter 5, however, this type of method can adequately represent only the variables. The objects are not represented. In chapter 5, INDOQUAL has been proposed as an alternative method for PCA of qualitative variables, which optimizes the sum of squared loadings, just as PCA does, but with respect to certain constraints in order to allow for a representation of the objects.

Neither of the methods MCA, PRINCIPALS, and INDOQUAL adequately takes into account *all* a method for PCA of qualitative variables should take into account. INDOQUAL differs essentially from MCA and PRINCIPALS, however, in that it does not first reduce the information of a qualitative variable by means of one or several optimal quantifications. INDOQUAL tries to take into account the variables as a whole, as is done in PCA of quantification matrices. The PCA objective has been relaxed in order to have component scores for the objects that are related to the components for the variables.

6.2. MCA as a method for finding an approximate solution for INDOQUAL

In the previous section a first difference in interpretation of MCA and INDOQUAL has been given, focusing on the idea that the methods should give a PCA representation of the variables. In the present section MCA is considered as a method for finding an approximation of the solution of INDSCAL on the quantification matrices. This gives another comparison of MCA and INDOQUAL.

As has been described earlier, INDORT is a constrained variant of INDSCAL (Carroll & Chang, 1970). INDSCAL minimizes the loss function

$$\sigma(X,W_1,...,W_m) = \sum_{j=1}^{m} \| S_j - XW_jX' \|^2, \quad (1)$$

over arbitrary X and diagonal matrices $W_1,...,W_m$. That is, INDSCAL fits the INDSCAL model

$$S_j \mathrel{\hat{=}} XW_jX', \quad (2)$$

for $j = 1,...,m$, to the data in the least squares sense, where $\hat{=}$ denotes a least squares approximation.

As mentioned by Carroll and Chang (1970), the INDSCAL model has been proposed earlier by Horan (1969). The latter, however, did not consider least squares fitting of this model. Instead, he proposed to find the object coordinates as follows. If the model fits the data perfectly, that is, if $S_j = XW_jX'$, for $j = 1,...,m$, then $\sum_j S_j = \sum_j XW_jX' = X(\sum_j W_j)X'$. It follows that, in that case, the object coordinates matrix is given by a transformation of the matrix containing the eigenvectors of $\sum_j S_j$, because these span the column- and row-spaces of $\sum_j S_j$. Hence the matrix with eigenvectors gives "the location of the points in the normal space" (Horan, 1969, p.144), where "normal space" refers to the space for the object coordinates. Horan realized that this solution is determined up to a linear transformation only. As Carroll and Chang (1970) point out, this undeterminacy is problematic in that it overlooks one of the key features of INDSCAL, that is, uniquely oriented axes. In addition, Carroll and Chang's procedure solves for the weight matrices W_j, whereas Horan ignores the estimation of the W_j matrices. Furthermore, Horan's approach is based on the assumption of perfect fit of the INDSCAL model, which does not hold in practice. Therefore, least squares fitting of the INDSCAL model, as done by Carroll and Chang (1970), is to be preferred.

Horan's approach, however, is of more than only historical interest. If the S_j matrices are the quantification matrices $P_j = JG_jD_j^{-1}G_j'J$, then Horan's approach comes down to MCA. That is, MCA can be seen as a method giving an approximate solution to INDSCAL applied to these same quantification matrices.

Because the MCA object coordinates matrix is column-wise orthonormal, this automatically provides an approximate solution for INDOQUAL (that is, INDORT applied to the P_j matrices) as well. This provides a first comparison between MCA and INDOQUAL. It is of interest to mention here that a similar result was found by Marchetti (1988) in comparing TUCKALS-3 for qualitative variables and Tucker's original approximate solution for the Tucker-3 model (Tucker, 1966).

Horan (1969) has not been the only one who used the eigenvectors of $\sum_j S_j$ as a representation of the INDSCAL object coordinates. Escofier and Pagès (1983) mention that the results of their Analyse Factorielle Multiple (AFM), which is also based on the eigenvectors of $\sum_j S_j$ and comes down to MCA when applied to quantification matrices $S_j = P_j = JG_jD_j^{-1}G_j'J$, can be interpreted as those of an INDSCAL analysis. Later on, Escofier and Pagès (1984) motivate this interpretation by considering their method as an alternative to the least squares fitting of the INDSCAL model. Instead of maximizing the *sum of squares* of certain projections, as INDSCAL does in their point of view, AFM maximizes the *sum* of these projections (nonsquared). That is, let $\mathbf{s}_j = \text{Vec}(S_j)$, where Vec(.) denotes the matrix between brackets strung out row-wise into a column, let w_{jl} be the l^{th} diagonal element of W_j, let $w_{1l},...,w_{ml}$ be collected in the vector $\mathbf{w}_j$, and let $\mathbf{x}_l$ be the l^{th} column of X. Then the INDSCAL model can be written as

$$\begin{aligned} \mathbf{s}_j &\mathrel{\hat{=}} \text{Vec}(\sum_{l=1}^{r} w_{jl}\mathbf{x}_l\mathbf{x}_l') \\ &= (\text{Vec}(\mathbf{x}_1\mathbf{x}_1')\,|\,...\,|\,\text{Vec}(\mathbf{x}_r\mathbf{x}_r'))\mathbf{w}_j. \end{aligned} \tag{3}$$

This model can be recognized as a multiple regression model. Finding $\mathbf{w}_j$ comes down to projecting $\mathbf{s}_j$ on the space spanned by the columns $\text{Vec}(\mathbf{x}_1\mathbf{x}_1'),...,\text{Vec}(\mathbf{x}_r\mathbf{x}_r')$. It can be stated that INDSCAL maximizes over X the sum of squared lengths of the projections of $\mathbf{s}_j$ on the column-space spanned by the vectors $\text{Vec}(\mathbf{x}_1\mathbf{x}_1'),...,\text{Vec}(\mathbf{x}_r\mathbf{x}_r')$.

On the other hand, AFM maximizes over column-wise orthonormal X the sum of the *nonsquared* projection lengths of the vectors $\mathbf{s}_j$ on each of the vectors $\text{Vec}(\mathbf{x}_1\mathbf{x}_1'),...,\text{Vec}(\mathbf{x}_r\mathbf{x}_r')$ *separately*. One may assume, without loss of generality, that the vectors $\mathbf{x}_1,...,\mathbf{x}_r$ are normalized to unit length. Hence the vectors $\text{Vec}(\mathbf{x}_1\mathbf{x}_1'),...,\text{Vec}(\mathbf{x}_r\mathbf{x}_r')$ have unit length also. Then the length of the projection of $\mathbf{s}_j$ on the vector Vec($\mathbf{x}_l\mathbf{x}_l'$) is given by $w_{jl} = \mathbf{s}_j'\text{Vec}(\mathbf{x}_l\mathbf{x}_l') = \mathbf{x}_l'S_j\mathbf{x}_l$. Thus AFM maximizes $\sum_j\sum_l w_{jl} = \mathbf{x}_l'S_j\mathbf{x}_l = \sum_j \text{tr}\, X'S_jX$.

Escofier and Pagès (1984) claim that the difference between AFM and INDSCAL is only a difference between maximizing a sum of squared projection lengths versus maximizing a sum of nonsquared projection lengths. There is an important additional difference between AFM and INDSCAL, however. In INDSCAL the sum of squared lengths of projections on the *subspace spanned by* the vectors Vec($\mathbf{x}_1\mathbf{x}_1'$),...,Vec($\mathbf{x}_r\mathbf{x}_r'$) is maximized. That is, apart from the difference of maximizing a sum of nonsquared versus squared projection lengths between AFM and INDSCAL, there is also a difference between the projections that are of interest in the different methods. In AFM projections on the *individual* axes are considered, whereas in INDSCAL projections on the *subspace* spanned by these axes are considered. This essential difference between INDSCAL and AFM seems to have been overlooked by Escofier and Pagès (1984). Maximizing either a sum or a sum of squares of lengths of projections on the *individual* axes (as AFM does) does not seem very interesting. When more than one axis is taken, the projection of a vector ($\mathbf{s}_j$) on the subspace as a whole should be considered for the approximation of the vector $\mathbf{s}_j$ by a vector in a subspace.

It is of interest to note that the orthonormally constrained variant of INDSCAL, IND***ORT*** can be interpreted in *both* of the above senses, that is, for INDORT the sum of squared lengths of projections on the individual axes is equal to the sum of squared lengths of the projections on the subspace spanned by the axes, due to the orthonormality of $\mathbf{x}_1,...,\mathbf{x}_r$, and hence of Vec($\mathbf{x}_1\mathbf{x}_1'$),...,Vec($\mathbf{x}_r\mathbf{x}_r'$). INDORT can hence be considered both as the method that maximizes the sum of squared lengths of projections on the *subspace*, and as the method that maximizes the sum of squared lengths of projections on the *individual axes*. It should be noted, however, that the orthonormality of the axes in AFM does not likewise guarantee that the sum of *nonsquared* lengths of projections on the individual axes is equal to the sum of *nonsquared* lengths of projections on the subspace spanned by these axes.

It can be concluded that Horan (1969) and Escofier and Pagès (1984) both propose to use the MCA coordinates as an approximate solution for the object coordinates for INDOQUAL, but that they have different motivations. In spite of their efforts to show the similarity in objectives of MCA and INDOQUAL, their papers served in fact to highlight the differences between the methods.

Nevertheless, Horan's observation that, when the INDSCAL model fits the data perfectly, the object coordinates matrix is a transformation of the matrix of eigenvectors of $\sum_j S_j$ is of considerable importance, and its implications will be discussed in the next section.

6.3. Equivalence of MCA and INDOQUAL when the INDORT model fits the quantification matrices perfectly

As has been mentioned in the previous section, when the INDSCAL model perfectly fits the matrices S_j, $j = 1,...,m$, one has

$$\sum_{j=1}^{m} S_j = X(\sum_{j=1}^{m} W_j)X'. \tag{4}$$

Now suppose that X in (4) is column–wise orthonormal. Then the matrices S_j are not only fitted perfectly by the INDSCAL model, but also by the INDORT model. When an eigendecomposition of $\sum_j S_j$ is given by $\sum_j S_j = K \Lambda K'$, then it follows from (4) that, assuming that the elements of $\sum_j W_j$ are different, $X = K$, and $\sum_j W_j = \Lambda$, up to a permutation. Hence, when the quantification matrices $P_j = JG_jD_j^{-1}G_j'J$ are fitted perfectly by the INDORT model (in r dimensions), then the object coordinates given by INDOQUAL are the same as those found by MCA (in r dimensions). It should be noted that the MCA object coordinates do not even differ a rotation from the INDOQUAL object coordinates, hence the MCA components themselves (and not a rotation of them) give the unique INDOQUAL components.

Of course, the above result in itself is useless in practice, because the INDORT model will, typically, not fit the quantification matrices in a reasonable number of dimensions. In fact, the INDORT model does not necessarily provide a perfect fit of the quantification matrices in the maximal number of dimensions. However, when the data do not fit the INDORT model perfectly, but merely to a great extent, the INDOQUAL solution and the MCA solution are still likely to be similar.

6.4. A comparison of MCA and INDOQUAL in terms of χ^2-distances

The last comparison to be made here between INDOQUAL and MCA is based on the following useful, though not prevalent interpretation of MCA, in terms of χ^2–distances (Benzécri et al., 1973, see Gifi, 1981, p. 134). That is, Benzécri et al. (1973) describe Correspondence Analysis (CA) as the technique that yields an approximation of so-called χ^2–distances between rows (or columns, which will be disregarded here) of a contingency table. Let the h^{th} and h'^{th} rows of a contingency table of order $r \times c$ be given by $\mathbf{r}_h$ and $\mathbf{r}_{h'}$, with elements r_{hg} and $r_{h'g}$, respectively, $g = 1,...,c$. Let the row-marginals of the contingency table be given by r_h, $h = 1,...,r$, and let the column-marginals be given by f_g, $g = 1,...,c$. Then, the χ^2–distance between two rows of a contingency table can be defined by

$$d_{\chi^2}(\mathbf{r}_h,\mathbf{r}_{h'}) = \sum_{g=1}^{c} \frac{(r_{hg}/r_h - r_{h'g}/r_{h'})^2}{f_g}, \tag{5}$$

(e.g., Gifi, p. 134), where the constant grand total n has been omitted from the usual definition. Heiser and Meulman (1983) have shown that CA approximates these χ^2–distances by means of a weighted Principal Coordinates Analysis (Gower, 1966). It is well-known (e.g., Lebart, Morineau & Tabard, 1977) that MCA can be seen as correspondence analysis of the superindicator matrix $G = (G_1 | ... | G_m)$. As has been shown by Meulman (1986, p. 87), MCA can be seen as a (nonweighted) Principal Coordinates Analysis on the χ^2–distances (which are equivalent to the Mahalanobis distances she mentions) defined between the rows of the super-indicator matrix. This Principal Coordinates Analysis first applies the Torgerson transformation to the matrix of distances (in the present case χ^2 distances), yielding $\sum_j JG_jD_j^{-1}G_j'J$, see Meulman (1986, p.87), where the constant m^{-1} has been dropped for convenience, and subsequently finds the best low-rank approximation of this matrix by minimizing

$$\sigma_1(X) = \| \sum_{j=1}^{m} JG_jD_j^{-1}G_j'J - XX' \|^2, \tag{6}$$

where X is an arbitrary matrix of order $n \times r$. It is well-known (Eckart & Young, 1936) that this minimization problem is solved by finding the eigendecomposition of $\sum_j JG_jD_j^{-1}G_j'J = K\Lambda K'$, and choosing $X = K_r\Lambda_r^{1/2}$, where the subscript r indicates that only the first r eigenvalues and eigenvectors are taken. Clearly, the problem of minimizing (6) over arbitrary X is equivalent to that of minimizing

$$\sigma_2(X,W) = \| \sum_{j=1}^{m} JG_jD_j^{-1}G_j'J - XWX' \|^2, \tag{7}$$

over diagonal matrices W, and column-wise orthonormal matrices X. This, in turn can be shown to be equivalent to minimizing

$$\sigma_3(X,W^*) = \sum_{j=1}^{m} \| JG_jD_j^{-1}G_j'J - XW^*X' \|^2, \tag{8}$$

over diagonal W^* and column-wise orthonormal X. Minimizing (8) can be interpreted in a similar way as minimizing (6) has been interpreted. That is, let the χ^2-distance based on variable j between objects i and i' be given by

$$d_j^2(i,i') = \sum_{g=1}^{m_j} \frac{(h_{igj} - h_{i'gj})^2}{f_g}, \tag{9}$$

where h_{igj} denotes the "score" of object i on the indicator variable for category g of variable j, and f_g is the frequency of category g of variable j. Then MCA can be seen as the method that performs a "simultaneous" Principal Coordinates Analysis on the χ^2-distances defined between the objects, for each of the variables. This is done by requiring that the object coordinates to be found be the same for all variables.

Now INDOQUAL can be seen as the method with almost the same objective. That is, INDOQUAL minimizes

$$\sigma(X,W_1,\ldots,W_m) = \sum_{j=1}^{m} \| JG_jD_j^{-1}G_j'J - XW_jX' \|^2, \tag{10}$$

over diagonal matrices $W_1,\ldots,W_m$, and over X subject to $X'X = I_r$. Both MCA and INDOQUAL can be seen as methods for simultaneous Principal Coordinates Analysis of the χ^2-distances between the objects, defined for all variables separately. They differ in the way they handle the simultaneousness of the two analyses. That is, in MCA the object coordinates are required to be the same for all variables, whereas in INDOQUAL the object coordinates may differ in column-scaling of the coordinates. This difference implies that INDOQUAL has a greater freedom in fitting the χ^2-distances, and hence will better approximate these distances for all variables jointly. This implies that

INDOQUAL may need fewer components to represent the distances between the objects than MCA does.

6.5. Discussion

In the previous sections some comparisons have been made between MCA and INDOQUAL. It has been emphasized which are the advantages of INDOQUAL over MCA. Obviously, MCA also has certain advantages over INDOQUAL. One of these is that the MCA solution is nested, whereas the solution of INDOQUAL is not. That is, the one–dimensional solution of INDOQUAL is not necessarily contained in the two–dimensional solution of INDOQUAL, the two–dimensional solution is not contained in the three–dimensional solution, etc. As a consequence, one cannot simply choose between an r and an r+1–dimensional INDOQUAL solution by looking at the extra dimension in the latter.

It may be seen as another advantage of MCA that it can be interpreted as an ordinary PCA of the indicator variables, that is, it gives optimal representations of the objects and the categories. The present study, however, focuses on methods that optimally represent the objects, the categories, and the variables simultaneously. Obviously, data analysis of qualitative variables may sometimes focus on the categories (and then MCA or variants of MCA seem most useful), while at other times it may focus on the variables, or on the objects, in which cases a PCA representation can be given by any of the methods described in the present study, and in particular by INDOQUAL.

Meulman (1986) has described MCA as a method for representing χ^2–distances between objects, and has offered an alternative to MCA which is better adjusted to this objective of representing distances. That is, her method aims at finding a representation of the objects in which the distances between the objects are approximated in the least squares sense. In a similar way, alternatives to INDOQUAL might be constructed. That is, one may use a method for least squares fitting of the distances given according to the INDSCAL model to the χ^2–distances between the objects, as defined by each of the variables. In doing so, however, one has to part with the objective of optimally representing the variables.

7. INDORT FOR A MIXTURE OF QUALITATIVE AND QUANTITATIVE VARIABLES (INDOMIX)

7.1. Introduction

In chapter 5 it has been mentioned that PCA of qualitative variables can be distinguished in two types, one focusing on the categories and the objects, the other on the variables. It has also been shown that INDOQUAL is a compromise between these two approaches in that it yields the best possible representation of the variables which also yields a representation of the objects. Ordinary PCA can be applied only to quantitative variables, and INDOQUAL can be applied only to data sets that consist exclusively of qualitative variables. For the exploratory analysis of mixtures of qualitative and quantitative variables a different approach is needed. Kiers (1988) has discussed some existing methods for such data and proposed a new method as well. The present chapter is based for a large part on this paper.

The exploratory analysis of a mixture of qualitative and quantitative variables seems to have received far less attention in the literature than the exploratory analysis of qualitative variables. Here three types of methods can be distinguished. The first type was proposed by Young, Takane and De Leeuw (1978), see also Tenenhaus (1977). Their method, PRINCIPALS, has already been mentioned in section 6.1 for the case where only qualitative variables are involved. In the case of a mixture of qualitative and quantitative variables each qualitative variable is "optimally quantified" by means of one quantitative variable, and an ordinary PCA is performed on the complete set of quantified qualitative variables and variables that were quantitative already. As has been explained in section 6.1, this method may work well when the qualitative variables can be fully captured by means of one quantified variable, for instance when the qualitative variables can be seen as "polytomized" quantitative variables. However, if this is not the case, much information is lost when each qualitative variable is replaced by only one quantitative variable. Because our attention goes to methods that take into account the possibility of more-dimensional information in qualitative variables, this type of methods is ignored in the present study.

The second type of methods generalizes Multiple Correspondence Analysis

(MCA) to the effect that it can handle mixtures of qualitative and quantitative variables. These generalizations of MCA have been proposed independently by many authors (De Leeuw, 1973; Escofier, 1979; Nishisato, 1980, pp.103–107; De Leeuw & Van Rijckevorsel, 1980). Although the methods slightly differ in the way in which quantitative variables are transformed, all methods essentially use the same approach to handle qualitative variables. That is, let G_j be the indicator matrix for variable j, when variable j is a qualitative variable, and let $\mathbf{h}_j$ be the vector of scores on variable j, when variable j is quantitative. Then all methods mentioned above can be described as PCA of the supermatrix containing the columns of the matrices $JG_jD_j^{-1/2}$ for qualitative variables and (transformations of) the vectors $\mathbf{h}_j$ for quantitative variables. The particular method that performs PCA of the supermatrix containing the columns of the matrices $JG_jD_j^{-1/2}$ for qualitative variables and the vectors of standard scores, $\mathbf{z}_j$, divided by $n^{1/2}$, for the quantitative variables will be denoted here as PCAMIX.

As has been mentioned in chapter 5, the very fact that MCA performs a PCA of the complete set of indicator variables for all qualitative variables causes it to yield a non–optimal representation of the qualitative variables. In fact, MCA yields an optimal representation of the *categories* of the qualitative variables, not of the variables themselves. Analogously, the generalizations of MCA for analyzing mixtures optimally represent only the categories of the qualitative variables, rather than the qualitative variables themselves, because they use the same approach for the qualitative variables as MCA does.

The third type of methods is PCA of quantification matrices for mixtures of variables. These methods have been discussed in section 4.2. Just as PCA of quantification matrices for qualitative variables, PCA of quantification matrices for mixtures does not provide a representation of the objects. On the other hand, as has been shown above, PCAMIX and its variants do not provide an optimal representation of the variables. In the present chapter it is shown that INDORT applied to certain quantification matrices yields a compromise between these two types of methods. Apart from this it is shown that this method has some interesting properties in certain special cases.

7.2. INDORT for the analysis of a mixture of qualitative and quantitative variables (INDOMIX)

In chapter 3 it has been mentioned that various quantifications can be chosen for qualitative and quantitative variables. In the present chapter the same quantification matrices are chosen for qualitative variables as in chapter 5. That is, if the j^{th} variable is a qualitative variable the quantification matrix chosen here is given by

$$P_j = JG_jD_j^{-1}G_j'J. \qquad (1)$$

If the j^{th} variable is a quantitative variable the quantification matrix chosen here is given by

$$Q_j = n^{-1}\mathbf{z}_j\mathbf{z}_j'. \qquad (2)$$

It should be noted that these quantification matrices differ from those chosen by Saporta (1976) in his method for PCA of quantification matrices only in that Saporta (1976) uses the normalized version of P_j. In the sequel, INDORT applied to quantification matrices S_j chosen as P_j or Q_j, $j = 1,...,m$, will be denoted as "INDOMIX" (**IND**scal with **O**rthonormality constraints applied to quantification matrices for **MIX**ed variables).

7.3. INDOMIX as a compromise between PCA of η^2-coefficients and PCAMIX

INDOMIX can be interpreted in a number of different ways. It will be shown here that it optimally represents the variables (as a PCA technique does) while retaining a clear link with the representation of the objects. More precisely, it will be shown that INDOQUAL is a compromise between one of the methods for PCA of quantification matrices for mixed variables, "PCA of η^2-coefficients", and PCAMIX. First, PCA of η^2-coefficients will be described.

The PCA of the quantification matrices S_j taken as P_j or Q_j can be considered as PCA of a certain "correlation matrix". For a pair of qualitative variables the "correlation" is defined as the ϕ^2-coefficient, for a mixed pair the η^2-coefficient is used, and for a pair of two quantitative variables the squared product-moment correlation is used. This method is called PCA of η^2-coefficients, named after the correlation-coefficient used for a pair of mixed variables.

PCA of η^2-coefficients maximizes the function

$$g(F_1,...,F_r) = \sum_{j=1}^{m} \sum_{l=1}^{r} (\text{tr } F_l'S_j)^2, \tag{3}$$

over the $n \times n$ matrices F_l, $l = 1,...,r$, representing "components" of the variables, subject to the constraint $\text{tr } F_l'F_{l'} = \delta_{ll'}$, where δ denotes the Kronecker symbol. As in section 5.3, $\text{tr } F_l'S_j$ can be considered as the loading of variable j on component l, and hence PCA of η^2-coefficients can be seen as the method that maximizes a sum of squared loadings.

INDOMIX is the method that maximizes (cf. chapter 5, formula (8))

$$f(X) = \sum_{j=1}^{m} \sum_{l=1}^{r} (\mathbf{x}_l'S_j\mathbf{x}_l)^2 = \sum_{j=1}^{m} \sum_{l=1}^{r} (\text{tr } \mathbf{x}_l\mathbf{x}_l'S_j)^2, \tag{4}$$

over X, subject to $X'X = I_r$, with S_j chosen as P_j or Q_j. As in section 5.3, maximizing $g(F_1,...,F_r)$ over $F_1,...,F_r$, subject to the constraint $\text{tr } F_l'F_{l'} = \delta_{ll'}$, for all pairs l and l', and to the additional constraint that $F_l = \mathbf{x}_l\mathbf{x}_l'$ is equivalent to maximizing (4) over X, subject to $X'X = I_r$. Hence INDOMIX can be interpreted as PCA of η^2-coefficients subject to the additional constraint that $F_l = \mathbf{x}_l\mathbf{x}_l'$.

As has been mentioned above, PCA of η^2-coefficients does not provide coordinates for the objects. An advantage of INDOMIX over PCA of η^2-coefficients is that it does yield coordinates for the objects. As has been shown in section 5.3, each component for the variables is directly and uniquely linked to a component for the objects.

It is well-known that the PCAMIX solution yields object coordinates for which $X'X = I_r$. That is, the PCAMIX solution satisfies the constraints imposed on the components in INDOMIX. INDOMIX yields the best possible representation of the variables, subject to these constraints. Therefore, INDOMIX yields a representation of the variables that is *better* than the one given by PCAMIX. This follows also from the fact that PCAMIX can be seen as a constrained variant of INDORT, because PCAMIX can be seen as applying SUMPCA to the quantification matrices S_j, which is a constrained variant of INDORT, as described in chapter 2.

It can be concluded that INDOMIX is a method that optimally represents relations among mixtures of variables, and also yields a representation of the objects. Clearly, in this way, INDOMIX is a compromise between PCA of η^2–coefficients and PCAMIX, in that it consists of a (constrained) PCA of the variables and simultaneously yields coordinates for the objects (like PCAMIX does). It should be noted that this interpretation also follows from the hierarchical relations discussed in section 4.2.

7.4. The interpretation of the results of an INDOMIX analysis

Because INDOMIX is a compromise between PCA of η^2–coefficients and PCAMIX, its results partly parallel those of PCA of η^2–coefficients and partly parallel those of PCAMIX. That is, like in PCAMIX, INDOMIX provides object coordinates, collected in matrix X. These can be interpreted in the same way as in PCAMIX, but PCAMIX and INDOMIX emphasize different aspects. That is, whereas PCAMIX emphasizes optimal representation of the objects and the categories, INDOMIX aims at optimal representation of the objects and the variables. As a consequence, PCAMIX does not provide an optimal representation of the variables, and INDOMIX does not provide an optimal representation of the categories. Nevertheless, it is possible to provide category coordinates for the INDOMIX solution, by computing $Y_j = D_j^{-1}G_j'X$, the matrix of centroids of the object coordinates of the objects that fall in the category concerned, for every category of a qualitative variable.

The results of INDOMIX share with the solution of PCA of η^2–coefficients that a representation of the variables is given. This representation is provided by the diagonal matrices $W_j = \text{Diag}(X'S_jX)$. The elements of these matrices can be interpreted as the loadings of the variables on the components. As far as qualitative variables are concerned, these loadings can again (as in chapter 5) be seen as η^2–coefficients, each with a maximum of 1. For a quantitative variable, the loadings on the components are squared product–moment correlations between the variable and the components concerned. In addition to these squared correlations it is useful to inspect the nonsquared correlations, and their signs.

Finally, we have an overall value for evaluating the quality of the solution. To this end, we use the proportion of inertia accounted for of the quantification matrices S_j. This proportion is given by the maximal value of f(X) (cf.(4)), divided by the total inertia. The total inertia of matrix S_j is equal to tr S_j^2, which is (m_j-1) for a qualitative variable and 1 for a quantitative variable. Let m_a be the number of qualitative variables, and m_b the number of quantitative variables, then the overall total inertia is equal to $(\sum_j m_j - m_a + m_b)$, where $\sum_j m_j$ is the number of categories of the qualitative variables. The proportion of inertia accounted for by the INDOMIX solution (IAF_I) is given by

$$\mathrm{IAF}_I = \frac{\sum_{j=1}^{m} \sum_{l=1}^{r} (\mathbf{x}_l' S_j \mathbf{x}_l)^2}{(\sum_j m_j - m_a + m_b)} . \tag{5}$$

In order to provide an indication of the quality of the INDOMIX solution, it is useful to compare this measure to the inertia of the quantification matrices that is accounted for by means of PCA of η^2-coefficients and PCAMIX, respectively. For PCA of η^2-coefficients, as in ordinary PCA, the proportion of inertia accounted for of the quantification matrices is given by the sum of the first r eigenvalues of the "correlation" matrix, divided by $(\sum_j m_j - m_a + m_b)$.

If the object coordinates are computed by means of PCAMIX, one might compute the proportion of inertia accounted for by means of (5) with the PCAMIX object coordinates substituted for the INDOMIX object coordinates. It should be noted, however, that the thus computed "proportion of inertia accounted for" is the inertia accounted for by the PCAMIX object coordinates when the quantification matrices are approximated by the INDORT model. Another interesting measure for the quality of the PCAMIX solution would be a measure that is based on the model for quantification matrices that is actually fitted by PCAMIX. In section 4.2, PCAMIX has been described as the method that applies SUMPCA to the quantification matrices S_j. That is, PCAMIX fits the quantification matrices to the model

$$\hat{S}_j = XWX', \tag{6}$$

for $j = 1,\ldots,m$, where W is a diagonal matrix, and X is the orthonormal matrix of object coordinates. It has been shown in chapter 2 that this model is a constrained variant of the INDORT model, with $W_j = W$, for all j. This model is interesting in itself, because, when it adequately represents the quantification matrices, it implies that all variables can be represented by the same coordinates in the variable space. Because for the PCAMIX solution we have $\sum_j \|S_j\|^2 = \sum_j \|\hat{S}_j - S_j\|^2 + \sum_j \|\hat{S}_j\|^2$, the inertia accounted for by the SUMPCA model (6), IAF$_S$, is expressed by

$$\text{IAF}_S = \frac{\sum\limits_{j=1}^{m} \text{tr } \hat{S}_j^{\,2}}{\sum\limits_{j=1}^{m} \text{tr } S_j^{\,2}} . \qquad (7)$$

The denominator of (7) is given by $\sum_j \text{tr } S_j^{\,2} = (\sum_j m_j - m_a + m_b)$. The numerator can be computed as follows. Obviously, $\text{tr } \hat{S}_j^{\,2} = \text{tr } XWX'XWX' = \text{tr } W^2$. From section 2.4 it readily follows that W is given by $m^{-1}\Lambda_r$, where Λ_r contains the first r eigenvalues of $\sum_j S_j$. Hence $\text{tr } \hat{S}_j^{\,2} = m^{-2} \sum_l \lambda_l^2$, where λ_l is the l^{th} eigenvalue of $\sum_j S_j$. Therefore,

$$\text{IAF}_S = \frac{\sum\limits_{j=1}^{m} \sum\limits_{l=1}^{r} \lambda_l^{\,2}}{m^2 (\sum_j m_j - m_a + m_b)} = \frac{\sum\limits_{l=1}^{r} \lambda_l^{\,2}}{m (\sum_j m_j - m_a + m_b)} . \qquad (8)$$

Comparing IAF$_I$ and IAF$_S$ provides the user with a tool to choose between representing the variables by means of INDOMIX and representing the variables by means of the simpler model with poorer fit, PCAMIX.

7.5. INDOMIX applied to sets of quantitative or dichotomous variables

Above, INDOMIX has been described as a method for the analysis of a mixture of variables. One special case of this method is the case where INDOMIX is applied to qualitative variables only. In that case INDOMIX is equivalent to INDOQUAL, described in chapter 5. Another interesting special case is the case where INDOMIX is applied to quantitative variables only. Apart from this special case the case where all variables are dichotomous will also be treated here, because it turns out to be a special case of INDORT applied to merely quantitative variables.

INDORT applied to a set of quantitative variables comes down to maximizing

$$\mathrm{f}(X) = \sum_{j=1}^{m} \sum_{l=1}^{r} (\mathbf{x}_l'S_j\mathbf{x}_l)^2 = \sum_{j=1}^{m} \sum_{l=1}^{r} (n^{-1}\mathbf{x}_l'\mathbf{z}_j\mathbf{z}_j'\mathbf{x}_l)^2. \qquad (9)$$

Clearly, $(n^{-1}\mathbf{x}_l'\mathbf{z}_j\mathbf{z}_j'\mathbf{x}_l)$ can be rewritten as $n^{-1}(\mathbf{z}_j'\mathbf{x}_l)^2$, which is the square of the loading a_{jl} of variable j on component l. Hence INDORT applied to a set of quantitative variables can be seen as the method that maximizes the sum of fourth powers of loadings of the variables on the components, $\sum_j\sum_l a_{jl}^4$, over X. Typically, this "PCA" will not yield the same solution as ordinary PCA of quantitative variables. In section 8.5 this method is discussed a little further.

It is well-known that MCA (and hence PCAMIX) applied to a set of dichotomous (also called "binary") variables can be seen as an ordinary PCA of the dichotomous variables when the scores (zero and one) are standardized, see, for instance, De Leeuw (1973, p.56–57). This property is explained here again, and it is shown that a similar property exists for INDOQUAL.

When all variables are dichotomous, the indicator matrix for the j^{th} variable can be described by $G_j = (\mathbf{h}_j|\mathbf{h}_{j'})$, where $\mathbf{h}_j$ is the vector containing the zero–one scores on the dichotomous variable j, and $\mathbf{h}_{j'} = \mathbf{1}-\mathbf{h}_j$. It is readily verified that $J\mathbf{h}_{j'} = -J\mathbf{h}_j$, hence $JG_j = (J\mathbf{h}_j|J\mathbf{h}_{j'}) = (J\mathbf{h}_j|-J\mathbf{h}_j)$. The matrix $D_j = G_j'G_j$ has diagonal elements f_j and $(n-f_j)$, where f_j is the frequency of the unit–elements in $\mathbf{h}_j$. Then $JG_jD_j^{-1}G_j'J = \left(f_j^{-1}+(n-f_j)^{-1}\right)J\mathbf{h}_j\mathbf{h}_j'J = nf_j^{-1}(n-f_j)^{-1}J\mathbf{h}_j\mathbf{h}_j'J$. It is well–known that the variance of a dichotomous variable j is given by $n^{-2}f_j(n-f_j)$, hence, when $\mathbf{z}_j$ denotes the standardized version of $\mathbf{h}_j$, we have $JG_jD_j^{-1}G_j'J = n^{-1}\mathbf{z}_j\mathbf{z}_j'$.

From the above it follows that MCA of dichotomous variables finds object coordinates as the (normalized) eigenvectors of $n^{-1}\sum_j\mathbf{z}_j\mathbf{z}_j'$. Clearly, these object coordinates are the same as the component scores found by PCA of a matrix Z, containing as columns the vectors $\mathbf{z}_j$. INDOMIX uses the quantification matrices $S_j = JG_jD_j^{-1}G_j'J$. In case the variables are dichotomous, these quantification matrices are given by $JG_jD_j^{-1}G_j'J = n^{-1}\mathbf{z}_j\mathbf{z}_j'$, hence $S_j = n^{-1}\mathbf{z}_j\mathbf{z}_j'$. It follows that, when INDOMIX is applied to dichotomous variables, the dichotomous variables can be considered as quantitative variables (with quantification matrices $n^{-1}\mathbf{z}_j\mathbf{z}_j'$). Furthermore, because INDOMIX maximizes $\sum_j\sum_l(\mathbf{x}_l'S_j\mathbf{x}_l)^2$ subject to $X'X = I_r$, it can be seen that INDOMIX applied to dichotomous variables maximizes $n^{-2}\Sigma_j\Sigma_l(\mathbf{x}_l'\mathbf{z}_j\mathbf{z}_j'\mathbf{x}_l)^2 = n^{-2}\Sigma_j\Sigma_l(\mathbf{z}_j'\mathbf{x}_l)^4$, where $n^{-1/2}\mathbf{z}_j'\mathbf{x}_l$ is the point–biserial correlation between variable j and component l. This point–biserial correlation can be considered as the loading of variable j on component l. It follows that INDORT for dichotomous variables maximizes the sum of fourth powers of the loadings of the variables on the components. However, with a different interpretation of the squared point–biserial correlation, that is, as an E–correlation coefficient based on the quantification matrices $n^{-1}\mathbf{z}_j\mathbf{z}_j'$ and $\mathbf{x}_l\mathbf{x}_l'$, the "loading" of variable j on component l is given by $\mathrm{tr}\, n^{-1}\mathbf{z}_j\mathbf{z}_j'\mathbf{x}_l\mathbf{x}_l' = n^{-1}(\mathbf{z}_j'\mathbf{x}_l)^2$, and hence INDORT for dichotomous variables can be seen as the method that maximizes the sum of squares of loadings of the variables on the components, as in ordinary PCA.

Above, it has been shown that dichotomous variables can be treated as quantitative variables by both PCAMIX and INDOMIX. This is very useful in practice, because the INDOMIX program can handle more quantitative variables than qualitative variables. However, when the INDOMIX solution is computed by treating dichotomous variables as quantitative variables, for these variables only the (point–biserial)–correlations are given, instead of the category coordinates. One can compute the category coordinates from these correlations as follows. Let y_{1jl} and y_{2jl} be the category centroids of the first and second categories, respectively, of variable j for component l. Because $\mathbf{h}_j'J\mathbf{h}_j = n^{-1}f_j(n-f_j)$, where f_j is the category frequency of the first category of variable j, and $\mathbf{x}_l = J\mathbf{x}_l$, y_{1jl} is given by $f_j^{-1}\mathbf{h}_j'\mathbf{x}_l = n^{-1/2}f_j^{-1/2}(n-f_j)^{1/2}(\mathbf{h}_j'J\mathbf{h}_j)^{-1/2}\mathbf{h}_j'J\mathbf{x}_l = n^{-1/2}f_j^{-1/2}(n-f_j)^{1/2}a_{jl}$, where a_{jl} is the point–biserial correlation between variable j and component l. Similarly, y_{2jl} can be shown to be equal to $-n^{-1/2}f_j^{1/2}(n-f_j)^{-1/2}a_{jl}$.

7.6. Discussion

In the present chapter INDOMIX has been described as a compromise between PCA of η^2–coefficients and PCAMIX. It is a compromise in that it yields a good representation of the variables (like PCA of η^2–coefficients) and at the same time it yields object coordinates (like PCAMIX). However, as has been shown in section 4.2, INDOMIX is not the only method that yields such a compromise. It has been shown there that TUCKALS–3 and unconstrained INDSCAL applied to the same quantification matrices yield other compromises between PCA of η^2–coefficients and PCAMIX.

In the present chapter a particular choice has been made for the quantification matrices for the qualitative and quantitative variables. However, as has been said in section 4.4, it is an open question whether other choices of quantification matrices might be more useful. An important consequence of the possibility of different choices for quantification matrices is that, apart from INDORT on other quantification matrices, also alternatives for PCAMIX and PCA of η^2–coefficients can be developed. That is, alternatives of PCAMIX can be developed as methods that find object coordinates as the first r eigenvectors of the sum of the (alternative) quantification matrices. Implicitly, such an alternative method has been proposed by Gower (1971), and it is actually used by Cuadras (1989). Some alternatives for PCA of η^2–coefficients have in fact been proposed by Janson and Vegelius (1978a, 1982), by representing the associations in a set of qualitative and quantitative variables by other generalized correlation coefficients than the ones chosen by Saporta (1976).

8. SIMPLE STRUCTURE IN COMPONENTS ANALYSIS FOR MIXTURES OF QUALITATIVE AND QUANTITATIVE VARIABLES

8.1. Introduction

The present chapter focuses on two main subjects. First, a procedure for simple structure rotation for PCAMIX is considered. Next, it is shown that INDOMIX is closely related to this simple structure rotation. This relation between INDOMIX and the simple structure rotation for PCAMIX provides a useful new interpretation of INDOMIX. It explains why the loadings of the variables obtained by INDOMIX are more clearly clustered than those of PCAMIX. In addition, it leads to a better understanding of a phenomenon observed with INDOQUAL. That is, INDOQUAL does not only yield clearer clusters of variables, it also tends to yield solutions with clusters of *objects* that, per component, are more clearly separated and denser than those found in an MCA solution. Apart from giving a formal explanation of this phenomenon, it will be illustrated by means of an example data set. First, however, it will be explained why simple structure rotation of the PCAMIX solution might be useful.

In ordinary PCA, that is, PCA of quantitative variables, the solution for the components (the object scores) and the loadings is determined up to a rotation only. The purpose of so-called "simple structure" rotation is to obtain components that have a clear interpretation in terms of subsets of the variables. Simple structure criteria are usually defined in terms of optimal patterns of (in absolute sense) small and large loadings. In general, techniques for rotation to simple structure "are concerned with attaining a factor matrix with a maximum tendency to have both small and large loadings" (Kaiser, 1958, p.188). For a detailed discussion on the rationale behind simple structure rotation the reader is referred to Harman (1976).

As has been explained in chapter 7, PCAMIX can be formulated as PCA of the total set of (binary) indicator variables supplemented with the quantitative variables. Therefore, the object coordinates can be seen as component scores just as in ordinary PCA. Moreover, as in ordinary PCA, the component scores are determined up to a rotation only. This is most easily verified by noting that PCAMIX maximizes the function $f(X) = \text{tr } X'\sum_j S_j X$, with S_j chosen as P_j or Q_j (as in chapter 7). Obviously, rotating the object coordinates matrix X by an orthonormal matrix T does not change the function value.

In practice, this rotational freedom seems not to have been used for finding "simple structure". In MCA (the special case of PCAMIX where all variables are qualitative) it is standard practice to use as components those that successively account for the maximum inertia, and ignore further rotations. In PCAMIX rotation does not seem to have been considered either. As is the case in ordinary PCA, the (unrotated) eigenvectors may yield components that are difficult to interpret. Therefore, the first purpose of the present chapter is to provide a method for rotating the component scores such that the best interpretable solution is found, according to some criterion.

As has been mentioned above, in ordinary PCA one rotates the component scores such that the loadings have optimal simple structure. That is, simple structure is expressed in terms of the loadings of the variables on the components. If one wants to optimize similar simple structure criteria by rotating the PCAMIX components, first of all one needs to define loadings, or rather squared loadings, of the variables on the PCAMIX components.

8.2. A definition of squared loadings in PCAMIX

Like PCA, PCAMIX finds component scores for the objects on several components. In ordinary PCA "loadings" of the variables on the components are given by the correlations between the variables and the components. In PCAMIX it is possible to define loadings for the quantitative variables in the same way. That is, the loading of the quantitative variable j on component l can be given by $a_{jl} = n^{-1/2}\mathbf{z}_j'\mathbf{x}_l$, the product-moment correlation between variable j and component l.

For the qualitative variables one cannot use the product-moment correlation. Instead, one has to choose another coefficient which expresses the correlation between a qualitative variable and a (quantitative) component. Such a measure in MCA is the "discrimination measure" (Gifi, 1981), which is the contribution of a component to the inertia of a variable accounted for. This discrimination measure is given by $c_{jl} \equiv \mathbf{x}_l'P_j\mathbf{x}_l$. Gifi (1981, p.96) explains that this measure can be seen as the squared correlation between variable j when it is "optimally quantified" and component l. Another interpretation of c_{jl} is that it is the well-known correlation ratio η^2. In both interpretations the measure c_{jl} is considered as a squared correlation. Therefore, it will be considered here as the squared loading of variable j on component l.

In order to have the same notation for qualitative and quantitative variables, c_{jl} is defined for a quantitative variable as $c_{jl} = a_{jl}^2 = n^{-1}(\mathbf{z}_j'\mathbf{x}_l)^2 = \mathbf{x}_l'(n^{-1}\mathbf{z}_j\mathbf{z}_j')\mathbf{x}_l$. Hence defining S_j as P_j or Q_j, we have $c_{jl} = \mathbf{x}_l'S_j\mathbf{x}_l$ for both qualitative and quantitative variables. It is of interest to note that PCAMIX can be formulated as the method that maximizes $\sum_{jl}c_{jl}$, with c_{jl} defined as above, over X, subject to $X'X = I_r$.

Having defined squared loadings for variables on PCAMIX components, we are in a position to consider criteria that measure simple structure of the loadings. Before considering simple structure criteria for PCAMIX, some well-known simple structure criteria that are used with ordinary PCA will be discussed.

8.3. Simple structure rotations for PCA

Kaiser (1958) has described several simple structure criteria, as well as procedures to optimize these criteria over orthogonal rotations of the loading matrix. Some of these have later been included in the orthomax family of orthogonal rotations (Jennrich, 1970, Crawford & Ferguson, 1970; see Clarkson & Jennrich, 1988). The criteria which are optimized by these rotation techniques will be discussed briefly in the present section.

The orthomax family of simple structure rotations for PCA can be described as the set of techniques that maximize the orthomax criterion (denoted by the acronym ORMAX). This criterion is expressed in terms of the squared loadings of the variables on the components. Let the loading for variable j on component l be given by a_{jl}, $j = 1,\ldots,m$, $l = 1,\ldots,r$. Then the ORMAX criterion is given by

$$\text{ORMAX} = \sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^4 - \frac{\gamma}{m}\sum_{l=1}^{r}(\sum_{j=1}^{m} a_{jl}^2)^2. \tag{1}$$

Although in principle γ can be any scalar, it is assumed here that $0 \leq \gamma \leq 1$. Several special choices of γ result in well–known simple structure criteria. That is, choosing $\gamma = 0$ yields the quartimax criterion (QMAX)

$$\text{QMAX} = \sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^4, \tag{2}$$

which has originally been proposed by Ferguson (1954). On the other hand, choosing $\gamma = 1$ yields the varimax criterion (VMAX) proposed by Kaiser (1958)

$$\text{VMAX} = \sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^4 - \frac{1}{m}\sum_{l=1}^{r}(\sum_{j=1}^{m} a_{jl}^2)^2. \tag{3}$$

In addition to quartimax and varimax, Kaiser (1958) described the following three simple structure criteria: Carroll (1953) proposed to minimize

$$\text{CROSSMIN} = \sum_{j=1}^{m}\sum_{k<l}^{r} a_{jk}^2 a_{jl}^2; \tag{4}$$

Neuhaus and Wrigley (1954) proposed to maximize the overall variance of squared loadings (OVERMAX)

$$\text{OVERMAX} = \sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^4 - \frac{1}{mr}(\sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^2)^2; \tag{5}$$

Saunders (1953) proposed to maximize the kurtosis of the total set of loadings combined with the set of loadings with reversed sign, which is proportional to

$$\text{KURTMAX} = \sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^4 \Big/ \frac{1}{mr}(\sum_{j=1}^{m}\sum_{l=1}^{r} a_{jl}^2)^2. \tag{6}$$

So far, only the simple structure criteria themselves have been discussed. The techniques for optimizing these criteria will now be discussed briefly. First of all, it should be noted that optimizing these criteria over orthogonal rotations of the component scores is equivalent to optimizing these criteria over orthogonal rotations of the loadings. This follows from the fact that, when the $m \times r$ matrix $A = n^{-1/2}Z'X$ contains the loadings of the variables (in matrix Z) on the components (in matrix X), a rotation of the components by a matrix T is paralleled by the same rotation of the matrix of loadings. That is, $AT = n^{-1/2}Z'XT$ contains the loadings of the variables on the rotated components. The techniques for optimizing the simple structure criteria mentioned above are all techniques for rotating the loading matrix.

In considering the different techniques for rotating the loading matrix, it is useful to note that the quartimax criterion and the varimax criterion are both special cases of the orthomax criterion. Moreover, as has been shown by Kaiser (1958), minimizing the CROSSMIN criterion and maximizing the OVERMAX and KURTMAX criteria over orthogonal rotations of the loading matrix is equivalent to maximizing the quartimax criterion. As a consequence, any of the optimization problems mentioned above can be subsumed under the general problem of maximizing the orthomax criterion over orthogonal rotations of the loading matrix. A description of maximizing the orthomax function, which turns out to be particularly useful in the present context (as will become clear later), was given by Ten Berge, Knol and Kiers (1988). Let A be the $m \times r$ matrix of loadings, and let $\mathbf{a}_j'$ be the j^{th} row of A. Ten Berge et al. (1988) define $E_j \equiv (\delta A'A - m\mathbf{a}_j\mathbf{a}_j')$, for $j = 1,...,m$, with δ defined such that $\gamma = \delta(2-\delta)$, and show that the problem of maximizing the orthomax function is equivalent to simultaneously diagonalizing the set of E_j matrices in the least squares sense, or equivalently, maximizing $\sum_j \text{tr}\,(\text{Diag}\,T'E_jT)^2$ over orthonormal matrices T. For this problem of simultaneously diagonalizing a set of matrices one can use an algorithm proposed by De Leeuw and Pruzansky (1978).

In the present section a number of simple structure criteria has been described and it has been pointed out that the orthogonal rotations that optimize these criteria can all be found by means of simultaneous diagonalization of the E_j matrices. In the next section, it will be shown how the same simple structure criteria can be optimized over rotations of the PCAMIX component scores solution.

8.4. Simple structure rotations for PCAMIX

In the present section, methods will be discussed for rotation of the

PCAMIX *component scores* such that the *loadings* have optimal simple structure in terms of the orthomax criteria. It should be noted that rotated component scores in PCAMIX do not correspond to a loading matrix which can be found by rotating the original loading matrix. This is a consequence of the fact that no definition of loadings for qualitative variables seems to be available such that rotating component scores corresponds to rotating the matrix of corresponding loadings. Therefore, in PCAMIX it does not suffice to express the simple structure criteria in terms of the original loadings and a rotation matrix. Instead, these criteria are to be expressed in terms of the squared loadings of the variables on the rotated components. It will now be shown how this can be accomplished.

The simple structure criteria given in (1), (2), (3), (4), (5), and (6) can be expressed in terms of the squared PCAMIX loadings by replacing a_{jl}^2 by c_{jl} for all j and l. The problem of optimizing the simple structure criteria over orthogonal rotations of the PCAMIX component scores will now be treated by giving an algorithm for maximizing the general ORMAX function (1) only. As has been explained above, the QMAX (2) and VMAX (3) functions are special cases of this function. Optimizing the CROSSMIN (4), OVERMAX (5), and KURTMAX (6) functions will be shown to be equivalent to maximizing the QMAX function.

The ORMAX criterion can be rewritten in terms of c_{jl} as

$$\text{ORMAX} = \sum_{j=1}^{m}\sum_{l=1}^{r} c_{jl}^2 - \frac{\gamma}{m}\sum_{l=1}^{r}(\sum_{j=1}^{m} c_{jl})^2. \tag{7}$$

In order to write the ORMAX criterion as a function f_{or} of the component scores matrix X the loading $c_{jl} = \mathbf{x}_l'S_j\mathbf{x}_l$ is substituted for c_{jl} in (7). This gives

$$\begin{aligned}\mathrm{f}_{or}(X) \equiv \text{ORMAX} &= \sum_{j=1}^{m}\sum_{l=1}^{r} (\mathbf{x}_l'S_j\mathbf{x}_l)^2 - \frac{\gamma}{m}\sum_{l=1}^{r}(\sum_{j=1}^{m} \mathbf{x}_l'S_j\mathbf{x}_l)^2\\ &= \sum_{j=1}^{m}\sum_{l=1}^{r} (\mathbf{x}_l'S_j\mathbf{x}_l)^2 - \frac{\gamma}{m}\sum_{l=1}^{r}(\mathbf{x}_l'\Sigma_jS_j\mathbf{x}_l)^2. \end{aligned}\tag{8}$$

In order to find the rotation that maximizes the ORMAX criterion we have to maximize $\mathrm{f}_{or}(FT)$ over orthonormal matrices T, where F contains the unrotated component scores solution for PCAMIX. The problem of finding the rotation that maximizes f_{or} will be translated here into a problem of simultaneously diagonalizing a set of matrices, just as has been done by Ten Berge et al. (1988) for the orthomax rotation of a loading matrix. Let the l^{th} column of T be given by $\mathbf{t}_l$, then $\mathrm{f}_{or}(FT)$ is given by

$$\begin{aligned}\mathrm{f}_{or}(FT) &= \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'F'S_jF\mathbf{t}_l)^2 - \frac{\gamma}{m}\sum_{l=1}^{r}(\ \Sigma_j\mathbf{t}_l'F'S_jF\mathbf{t}_l)^2 \\ &= \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'(F'S_jF-\delta m^{-1}\Sigma_kF'S_kF)\mathbf{t}_l)^2, \qquad (9)\end{aligned}$$

with δ chosen such that $2\delta-\delta^2=\gamma$. For $\delta=\gamma=1$, the right-hand side of (9) follows at once from the fact that the variance can be written as an average squared deviation from the mean. If $\delta \neq 1$, the second equality in (9) follows from the fact that

$$\begin{aligned}\sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'(F'S_jF-\delta m^{-1}\Sigma_kF'S_kF)\mathbf{t}_l)^2 &= \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'F'S_jF\mathbf{t}_l)^2 + \\ &\sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'\delta m^{-1}\Sigma_kF'S_kF\mathbf{t}_l)^2 - 2\sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'F'S_jF\mathbf{t}_l)(\mathbf{t}_l'\delta m^{-1}\Sigma_kF'S_kF\mathbf{t}_l) \\ = \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'F'S_jF\mathbf{t}_l)^2 + m\sum_{l=1}^{r}\delta^2m^{-2}(\mathbf{t}_l'\Sigma_jF'S_jF\mathbf{t}_l)^2 & \\ &- 2\ \delta m^{-1}\sum_{l=1}^{r}(\mathbf{t}_l'F'\Sigma_jS_jF\mathbf{t}_l)(\mathbf{t}_l'\Sigma_kF'S_kF\mathbf{t}_l) \\ = \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'F'S_jF\mathbf{t}_l)^2 + \delta^2m^{-1}\sum_{l=1}^{r}(\mathbf{t}_l'\Sigma_jF'S_jF\mathbf{t}_l)^2 &- 2\ \delta m^{-1}\sum_{l=1}^{r}(\mathbf{t}_l'\Sigma_jF'S_jF\mathbf{t}_l)^2 \\ = \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'F'S_jF\mathbf{t}_l)^2 + m^{-1}(\delta^2-2\delta)\sum_{l=1}^{r}(\mathbf{t}_l'\Sigma_jF'S_jF\mathbf{t}_l)^2. & \qquad (10)\end{aligned}$$

The columns of F are eigenvectors of Σ_jS_j, normalized to unit sums of squares, hence $F'\Sigma_jS_jF=\Lambda$, where Λ is the diagonal matrix with the first r eigenvalues of Σ_jS_j on its diagonal. With this result (9) can be rewritten as

$$\mathrm{f}_{or}(FT) = \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{t}_l'(F'S_jF-\delta m^{-1}\Lambda)\mathbf{t}_l)^2. \qquad (11)$$

Let $\tilde{E}_j$ be defined by

$$\tilde{E}_j \equiv (mF'S_jF - \delta\Lambda), \qquad (12)$$

for $j = 1,\ldots,m$. Then (11) can be rewritten as

$$\mathrm{f}_{or}(FT) = m^{-2} \sum_{j=1}^{m} \mathrm{tr}\ T'\tilde{E}_jT(\mathrm{Diag}\ T'\tilde{E}_jT). \qquad (13)$$

Ten Berge (1984, p.348) has shown that maximizing such a function over orthonormal matrices T is equivalent to the problem of simultaneously diagonalizing a set of matrices $\tilde{E}_1,\ldots,\tilde{E}_m$ in the least squares sense, and that hence the algorithm for this problem, proposed by De Leeuw and Pruzansky (1978), can be used, or any other algorithm for the simultaneous diagonalization of a set of symmetric matrices. Hence maximizing the orthomax criterion by rotating the PCAMIX component scores can be done by means of an algorithm for simultaneous diagonalization of matrices $\tilde{E}_1,\ldots,\tilde{E}_m$ with $\tilde{E}_j$ defined as in (12).

As has been mentioned above, Ten Berge et al. (1988) have shown that the problem of maximizing the orthomax function over orthogonal rotations of a PCA loading matrix is equivalent to the problem of simultaneously diagonalizing the matrices $E_j = (\delta A'A - m\mathbf{a}_j\mathbf{a}_j')$, where A is the $m \times r$ PCA loading matrix, and $\mathbf{a}_j'$ is the j^{th} row of A. It will now be shown that, in the special case where PCAMIX is applied to a set of merely quantitative variables, the procedure for the orthomax rotation of the PCAMIX solution can be seen as the simultaneous diagonalization of a set of $\tilde{E}_j$ matrices which are proportional to the E_j matrices in the Ten Berge et al. (1988) procedure. When all variables are quantitative the S_j matrices are given by $S_j = n^{-1}\mathbf{z}_j\mathbf{z}_j'$. Hence $\tilde{E}_j = (mn^{-1}F'\mathbf{z}_j\mathbf{z}_j'F - \delta\Lambda)$, where Λ is the diagonal matrix with eigenvalues of $n^{-1}\sum_j\mathbf{z}_j\mathbf{z}_j'$. Clearly, $\mathbf{a}_j' = n^{-1/2}\mathbf{z}_j'F$, hence $mn^{-1}F'\mathbf{z}_j\mathbf{z}_j'F = m\mathbf{a}_j\mathbf{a}_j'$. In addition, matrix $A'A = F'n^{-1}\sum_j\mathbf{z}_j\mathbf{z}_j'F = \Lambda$. Therefore, $\tilde{E}_j$ can be written as $\tilde{E}_j = (m\mathbf{a}_j\mathbf{a}_j' - \delta A'A) = -E_j$. Hence the simultaneous diagonalization of the matrices $\tilde{E}_j$ and that of the matrices E_j yield the same rotation matrix. It can be concluded that the orthomax rotation procedure for ordinary PCA is a special case of the orthomax rotation procedure for PCAMIX that is suggested here.

The quartimax criterion and the varimax criterion are special cases of the orthomax criterion. Therefore, with the orthomax rotation procedure for PCAMIX suggested above we have at once a quartimax and a varimax procedure for rotating the PCAMIX solution. These are given by setting γ to 0 or 1, respectively. The orthomax rotation procedure for PCAMIX has been described as the simultaneous diagonalization of a set of $\tilde{E}_j$ matrices as defined by (12). The $\tilde{E}_j$ matrices do not explicitly contain γ, but depend on γ because δ depends on γ. The $\tilde{E}_j$ matrices for the quartimax procedure are given by taking $\delta = 0$ (or $\delta = 2$, which is less convenient and therefore ignored), because then $\gamma = 0$. For the varimax procedure the $\tilde{E}_j$ matrices are given by $\delta = 1$, because then $\gamma = 1$.

So far only the QMAX and VMAX criteria have been considered. As has been shown by Kaiser (1958), for ordinary PCA component scores the procedure for optimizing the CROSSMIN, OVERMAX and KURTMAX criteria over orthogonal rotations are all equivalent to the quartimax rotation procedures. That is, these criteria are all optimized by the same orthogonal rotation matrix. For orthogonal rotation of the PCAMIX component scores again the same rotation matrix minimizes CROSSMIN and maximizes QMAX, OVERMAX and KURTMAX, as is shown below.

The CROSSMIN criterion for PCAMIX can be rewritten in terms of the squared loadings of the variables on the (rotated) components as

$$\text{CROSSMIN} = \sum_{j=1}^{m} \sum_{k<l}^{r} c_{jk}c_{jl} = \frac{1}{2} \sum_{j=1}^{m} \left(\sum_{l=1}^{r} c_{jl}\right)^2 - \frac{1}{2} \sum_{j=1}^{m} \sum_{l=1}^{r} c_{jl}^2. \qquad (14)$$

The first term in the right–hand side contains $\sum_l c_{jl} = \sum_l \mathbf{t}_l' F' S_j F \mathbf{t}_l =$ tr $T'F'S_jFT =$ tr $F'S_jF$, which does not depend on T, because T is orthonormal. Therefore, minimizing the CROSSMIN criterion over orthogonal rotations is equivalent to maximizing the second term in the right–hand side of (14). This term is the QMAX criterion (multiplied by 1/2) expressed in squared loadings c_{jl}. Therefore, minimizing the CROSSMIN criterion is equivalent to maximizing QMAX, for PCAMIX.

The OVERMAX and the KURTMAX criteria can be rewritten in terms of squared loadings c_{jl} as

$$\text{OVERMAX} = \sum_{j=1}^{m} \sum_{l=1}^{r} c_{jl}^{2} - \frac{1}{mr}\Big(\sum_{j=1}^{m} \sum_{l=1}^{r} c_{jl}\Big)^{2} \tag{15}$$

and

$$\text{KURTMAX} = \sum_{j=1}^{m} \sum_{l=1}^{r} c_{jl}^{2} \Big/ \frac{1}{mr}\Big(\sum_{j=1}^{m} \sum_{l=1}^{r} c_{jl}\Big)^{2}, \tag{16}$$

respectively. Both criteria contain the term $\sum_l c_{jl}$, which does not depend on the orthonormal rotation matrix T. As a consequence, finding the orthogonal rotation that maximizes these criteria depends only on the term $\sum_j \sum_l c_{jl}^{2}$, that is, again, the QMAX criterion. It can be concluded that finding the rotation matrix that optimizes the CROSSMIN, OVERMAX, and KURTMAX criteria for PCAMIX comes down to finding the orthogonal rotation that maximizes the QMAX criterion. The procedure for finding this rotation matrix has been discussed above as a special case of the procedure for orthomax rotation of the PCAMIX solution.

This concludes the discussion of simple structure rotation techniques for PCAMIX solutions. It is in no way intended to give a complete account of possible simple structure rotations for PCAMIX solutions. There are many other simple structure criteria for ordinary PCA, among which oblique simple structure rotations, as described, for instance by Clarkson and Jennrich (1988), that might be of interest for PCAMIX.

In the present section methods have been discussed for optimizing simple structure criteria for loadings of variables on PCAMIX component scores. These methods are based on rotation of the PCAMIX component scores. Such a rotation does not affect the optimality of the function that is maximized by PCAMIX, that is, $\sum_j \sum_l c_{jl}$, which can be seen as a measure for explained inertia. One might, however, want to find those components that maximize the simple structure criteria, possibly loosing the optimality of the function maximized by PCAMIX. In the case of merely quantitative variables, for instance, one might seek the components that have the maximal varimax or quartimax function value over all possible sets of orthogonal components, regardless of the variance they explain. In the next section, methods will be discussed for finding such components, that is, components that maximize the orthomax criterion for PCAMIX loadings over all possible sets of orthogonal components.

8.5. INDOMIX and a generalization

PCAMIX is the best-known method for the analysis of mixtures of variables. In chapter 7, however, an alternative method, INDOMIX, has been developed as a compromise between PCAMIX and "PCA of quantification matrices", in that it is directed at optimally representing the variables (as PCA of quantification matrices does) while at the same time providing component scores for the objects (as PCAMIX does). In the present section it is shown that this method optimizes the quartimax criterion over all possible sets of orthogonal component scores. In addition, an alternative method is discussed, which might be used for optimizing the varimax criterion, or any other criterion that belongs to the orthomax family.

INDOMIX comes down to maximizing

$$g(X) = \sum_{j=1}^{m} \text{tr } \text{Diag}(X'S_jX)^2 = \sum_{j=1}^{m}\sum_{l=1}^{r} (\mathbf{x}_l'S_j\mathbf{x}_l)^2 = \sum_{j=1}^{m}\sum_{l=1}^{r} c_{jl}^{\,2}, \qquad (17)$$

subject to $X'X = I_r$.

Clearly, maximizing g(X) is equivalent to maximizing the quartimax function. INDOMIX maximizes the quartimax function over orthogonal component scores matrices X, whereas the quartimax rotation applied to PCAMIX maximizes the quartimax function over orthogonally rotated component scores matrices FT. The class of matrices X that are only constrained by $X'X = I_r$ contains not only all rotations FT of F, but also matrices with columns outside the column-space of F. Therefore, the maximum of g(X) is always at least as large as the maximum of g(FT). This provides another interpretation of INDOMIX. INDOMIX maximizes the quartimax criterion over all possible orthogonal component scores matrices, in this way yielding a quartimax value that is always at least as high as the maximum possible quartimax value that can be obtained by orthogonal rotation of the PCAMIX solution. This implies that INDOMIX yields solutions that have a higher amount of simple structure than the optimally rotated PCAMIX solutions, when simple structure is defined in the quartimax sense.

The quartimax criterion being one of the first analytic simple structure criteria, it is not the most prevalent criterion in ordinary PCA. As Kaiser (1958) pointed out, using the quartimax criterion tends to yield solutions with one general component, which is quite contrary to the purpose of achieving maximum simple structure. This tendency of yielding a general component does not seem to be present with INDOMIX, and in practice it is found that, even though INDOMIX maximizes only the quartimax function, INDOMIX tends to yield simple structure in terms of other criteria (like varimax) as well, as will be discussed later. Nevertheless, INDOMIX does not maximize these other criteria, and it seems useful to discuss methods that do maximize other simple structure criteria (in the orthomax family) over all possible sets of orthogonal component scores, in the same way as INDOMIX maximizes the quartimax criterion.

The orthomax function (8) can be rewritten as

$$\begin{aligned}\mathrm{f}_{or}(X) &= \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{x}_l'S_j\mathbf{x}_l)^2 - \frac{\gamma}{m}\sum_{l=1}^{r}(\mathbf{x}_l'\Sigma_jS_j\mathbf{x}_l)^2 \\ &= \sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{x}_l'(S_j-\delta m^{-1}\Sigma_kS_k)\mathbf{x}_l)^2, \qquad (18)\end{aligned}$$

with δ again chosen such that $2\delta-\delta^2=\gamma$. The last step in the derivation of (18) is based on a similar reasoning as is made in deriving (10). Clearly, choosing $\delta=0$ (or $\delta=2$) we have $\mathrm{f}_{or}(X)=\mathrm{g}(X)$. However, (18) can be defined for any other γ between 0 and 1 (to which corresponds $\delta=1\pm(1-\gamma)^{1/2}$). In particular, the varimax function is maximized over all orthogonal component scores matrices X by maximizing

$$\mathrm{h}(X)=\sum_{j=1}^{m}\sum_{l=1}^{r}(\mathbf{x}_l'(S_j-m^{-1}\Sigma_kS_k)\mathbf{x}_l)^2, \qquad (19)$$

over X, subject to $X'X=I_r$. That is, this method applies INDORT to the matrices $(S_j-m^{-1}\Sigma_kS_k)$, which are the matrices S_j "centered" with respect to their mean.

Ten Berge et al. (1988) have provided an algorithm for maximizing (17). They have shown that this algorithm converges monotonically, when the S_j are positive semi-definite. Their algorithm might be used for maximizing (18) as well, with S_j replaced by $(S_j-\delta m^{-1}\Sigma_kS_k)$, but then monotone convergence of the algorithm is no longer guaranteed. An algorithm for which monotone convergence is guaranteed has been described by Kiers (in press). It is a slightly adapted version of the Ten Berge et al. algorithm.

It can be concluded that several methods are now available for finding components for mixtures of variables such that the orthomax simple structure criteria (for the loadings of the variables on the components) are maximized. One of these approaches is to perform PCAMIX first and then rotate the component scores such that they optimize the simple structure criteria over orthogonal rotations of the PCAMIX component scores. This procedure finds those sets of components that account best for the inertia as measured in PCAMIX, and, among these components, it finds that set of components that yields the highest simple structure value. The other approach, the generalization of INDOMIX that maximizes (18), is to seek those components that have the best possible simple structure, at the cost of a loss (which tends to be rather small in practice) in explained inertia.

The special case where the generalization of INDOMIX that maximizes the varimax function h(X) is applied to a set of merely quantitative variables is of special interest, because it provides an alternative to ordinary PCA. This method will yield varimax values that are always at least as high as the varimax values of rotated PCA loadings. Therefore, when one's main objective is to find components that have clear simple structure, that is, give clear clusters of variables, and when accounting for the variance is less important, then the generalization of INDOMIX might be a useful alternative to ordinary PCA. Moreover, although the explained variance $\sum_j\sum_l c_{jl}$ is no longer maximized by the generalization of INDOMIX, it cannot be very small either, because, this would contradict the maximality of h(X) which can be written as $\sum_j\sum_l(c_{jl} - m^{-1}\sum_k c_{kl})^2$, that is, the sum of column–variances of the elements of C.

In the present section, a method has been proposed for maximizing the simple structure criteria in the orthomax family, including the VMAX and QMAX criteria. In addition, maximizing the QMAX criterion over orthogonal rotations has been shown to be equivalent to optimizing the CROSSMIN, OVERMAX, and KURTMAX criteria, which are in this way also related to the orthomax family. However, when the orthomax function is maximized over all possible sets of orthogonal component scores, the equivalence between maximizing QMAX and optimizing CROSSMIN, OVERMAX, and KURTMAX can no longer be shown to hold. Nevertheless, there are some relations between INDOMIX and optimally rotated PCAMIX solutions in terms of these criteria. These and some other results are given in the next section.

8.6. Relations between INDOMIX and simple structure rotations of PCAMIX

In the previous section, the generalization of INDOMIX that maximizes the orthomax function over all possible orthogonal component scores matrices has been discussed. Obviously, this method always yields an orthomax function value that is at least as high as the one obtained by the orthomax rotation of the PCAMIX solution. As has been mentioned above, this method is a generalization of INDOMIX (chapter 7) which maximizes the quartimax function. In the present section, it will be shown that the latter method, INDOMIX, does not only yield a quartimax function value which is at least as high as the one attained by quartimax rotation of the PCAMIX solution, but that it also yields values at other simple structure criteria that are at least as high as the ones attained by optimally rotated PCAMIX solutions. These comparisons are summarized in Results 1 to 5. In these results the component scores matrix of the INDOMIX solution is denoted as X_I, that of the unrotated PCAMIX solution as F, and that of the optimally rotated PCAMIX solution as FT_0. That is, T_0 is the rotation matrix that optimizes the simple structure criterion at hand. The simple structure criteria given above as ORMAX, QMAX, VMAX, OVERMAX and KURTMAX are seen here as functions of the component scores matrices.

<u>Result 1</u>. QMAX(X_I) $\geq$ QMAX(FT_0) $\geq$ QMAX(F).

<u>Result 2</u>. a. OVERMAX(X_I) $\geq$ OVERMAX(FT_0) $\geq$ OVERMAX(F).

b. KURTMAX(X_I) $\geq$ KURTMAX(FT_0) $\geq$ KURTMAX(F).

<u>Result 3</u>. ORMAX(X_I) $\geq$ ORMAX(F).

<u>Result 4</u>. VMAX(X_I) $\geq$ VMAX(F).

<u>Result 5</u>. ORMAX(X_I) $\geq$ ORMAX(FT_0) $=$ ORMAX(F), if $m = 2$, and both variables are qualitative, that is, in case PCAMIX is equivalent to correspondence analysis.

The results given above are proven as follows.

Proof 1. From the fact that INDOMIX maximizes QMAX over all possible component scores matrices it follows at once that QMAX(X_I) $\geq$ QMAX(FT_O). The optimality of T_O guarantees that QMAX(FT_O) $\geq$ QMAX(F).□

Proof 2. The OVERMAX function can be written as

$$\text{OVERMAX}(X) = \text{g}(X) - m^{-1}r^{-1}\text{f}^2(X), \tag{20}$$

where f(X) is the function maximized by PCAMIX, and g(X) is the function maximized by INDOMIX (17). Obviously, f(FT_O) = f(F) $\geq$ f(X_I), and g(X_I) $\geq$ g(FT_O). Because f(X) is nonnegative for all X, it follows that $\text{f}^2(FT_O) \geq \text{f}^2(X_I)$. As a consequence $\left[\text{g}(X_I) - m^{-1}r^{-1}\text{f}^2(X_I)\right] \geq \left[\text{g}(FT_O) - m^{-1}r^{-1}\text{f}^2(FT_O)\right]$, that is OVERMAX($X_I$) $\geq$ OVERMAX(FT_O). Obviously, the optimality of T_O guarantees that OVERMAX(FT_O) $\geq$ OVERMAX(F). This completes the proof of Result 2a. For the proof of Result 2b it is useful to note that KURTMAX(X) = $mr\text{g}(X)/\text{f}^2(X)$, from which Results 2b follow at once, according to a similar reasoning as the one that is used in proving 2a. □

Proof 3. The ORMAX function (8) can be rewritten as

$$\text{ORMAX}(X) = \text{g}(X) - \frac{\gamma}{m}\,\text{k}(X), \tag{21}$$

where $0 \leq \gamma \leq 1$, and k(X) is defined as

$$\text{k}(X) \equiv \sum_{l=1}^{r} (\mathbf{x}_l'\textstyle\sum_j S_j \mathbf{x}_l)^2. \tag{22}$$

Let an eigendecomposition of $\sum_j S_j$ be given by $\sum_j S_j = K\Lambda K'$. Then k(X) can be rewritten as

$$\text{k}(X) = \sum_{l=1}^{r} (\mathbf{x}_l'K\Lambda K'\mathbf{x}_l)^2. \tag{23}$$

From the Cauchy–Schwarz theorem it follows that

$$(\mathbf{x}_l'K\Lambda K'\mathbf{x}_l)^2 = \left((\mathbf{x}_l'K\Lambda K')(\mathbf{x}_l)\right)^2 \leq (\mathbf{x}_l'K\Lambda^2K'\mathbf{x}_l) \tag{24}$$

and hence

$$\sum_{l=1}^{r} (\mathbf{x}_l'K\Lambda K'\mathbf{x}_l)^2 \;\le\; \sum_{l=1}^{r} (\mathbf{x}_l'K\Lambda^2K'\mathbf{x}_l) = \text{tr } X'K\Lambda^2K'X. \tag{25}$$

It is readily verified (Ten Berge, 1983) that the right-hand side in (25) is smaller than or equal to the sum of the first r diagonal elements of Λ^2. Let Λ_r denote the diagonal matrix containing the first r diagonal elements of Λ, then

$$\text{k}(X) = \sum_{l=1}^{r} (\mathbf{x}_l'K\Lambda K'\mathbf{x}_l)^2 \;\le\; \text{tr } \Lambda_r^{\,2}. \tag{26}$$

Inequality (26) yields an upper bound to $\text{k}(X)$. Clearly, this upper bound is attained by choosing X as K_r, the matrix with the r eigenvectors of $\sum_j S_j$ that belong to the first r eigenvalues of $\sum_j S_j$. This is precisely the unrotated PCAMIX solution F for the component scores. Therefore, $\text{k}(X_I) \le \text{k}(F)$, and, because $\gamma \ge 0$, $-\gamma\text{k}(X_I) \ge -\gamma\text{k}(F)$. Combining this result with the fact that $\text{g}(X_I) \ge \text{g}(F)$ proves that $\text{ORMAX}(X_I) \ge \text{ORMAX}(F)$. □

Proof 4. Result 4 follows immediately from Result 3 when γ is taken equal to 1.□

Proof 5. If $m = 2$ and both variables are qualitative it can be shown that $F'S_1F = F'S_2F = \frac{1}{2}\Lambda_r$. Hence $\text{f}_{or}(FT_0)$ is the maximum over T of

$$\text{f}_{or}(FT) = \sum_{j=1}^{2} \sum_{l=1}^{r} (\mathbf{t}_l'(F'S_jF - \delta\tfrac{1}{2}\Lambda_r)\mathbf{t}_l)^2, \tag{27}$$

as follows from (11). Substituting $F'S_1F = F'S_2F = \frac{1}{2}\Lambda_r$ in (27) yields

$$\text{f}_{or}(FT) = \sum_{j=1}^{2} \sum_{l=1}^{r} (\mathbf{t}_l'((1-\delta)\tfrac{1}{2}\Lambda_r)\mathbf{t}_l)^2. \tag{28}$$

It is proven analogously to the proof of (26) that

$$\text{f}_{or}(FT) = \sum_{j=1}^{2} \sum_{l=1}^{r} (\mathbf{t}_l'((1-\delta)\tfrac{1}{2}\Lambda_r)\mathbf{t}_l)^2 \;\le\; \sum_{j=1}^{2} \tfrac{1}{4}(1-\delta)^2 \text{tr } \Lambda_r^{\,2}. \tag{29}$$

The right-hand side of (29) gives an upper bound to $\text{f}_{or}(FT)$, which is attained for $T = I$. That is, $\text{f}_{or}(FT_0)$, the maximum over T of $\text{f}_{or}(FT)$ is equal to

$\mathrm{f}_{or}(F)$, hence ORMAX(FT_{O}) = ORMAX(F). With this equality ORMAX(X_I) $\geq$ ORMAX(FT_{O}) = ORMAX(F) follows immediately from Result 3.□

It is of interest to mention that, for $m = 2$, Result 5 implies that VMAX(F) = VMAX(FT_{O}) = 0. The fact that VMAX(FT_{O}) = 0 follows at once upon substitution of $\delta = 1$ in (29). That is, the Correspondence Analysis (MCA with $m = 2$) solution yields a varimax function which is always zero, and cannot be increased by rotating the solution.

Result 5 has been included, because it proves that VMAX(X_I) $\geq$ VMAX(FT_{O}) in a special case. This result does not generally hold for $m > 2$. Yet, in practice, it is often found that VMAX(X_I) $\geq$ VMAX(FT_{O}). This can be explained by the fact that VMAX = $\sum_j\sum_l c_{jl}^2 - m^{-1}\sum_l(\sum_j c_{jl})^2$, in which the first term is maximized by INDOMIX. The PCAMIX solution (and any rotation of this) maximizes $\sum_j\sum_l c_{jl}$. Obviously, a low value for $\sum_l(\sum_j c_{jl})^2$ for a certain PCAMIX solution would contradict the optimality of $\sum_j\sum_l c_{jl}$. Hence PCAMIX tends to find loadings for which $\sum_l(\sum_j c_{jl})^2$ is high. This explains why there is a tendency for the INDOMIX solution to yield a higher value of the VMAX criterion than any rotation of the PCAMIX solution. Along with Results 1 and 2, this result is of special interest when the INDOMIX and PCAMIX solutions are to be compared in terms of simple structure. In the next section, a comparison is made of the special cases of INDOMIX and PCAMIX where they are applied to sets of qualitative variables, that is, INDOQUAL and MCA, respectively. It will be shown that the higher simple structure values attained by INDOQUAL can be interpreted in terms of a better discriminatory capability of INDOQUAL compared to MCA, at the cost of a loss of explained inertia.

8.7. A comparison of MCA and INDOQUAL with respect to discriminatory capability

Above, it has been shown that INDOMIX attains values of several simple structure criteria which are at least as high as those attained by optimally rotated PCAMIX solutions. The present section discusses a consequence of this result for the case where only qualitative variables are involved. It has been mentioned by Van der Burg (1988, p.171 ff) that MCA can be seen as a cluster technique. It will be shown here that MCA (that is PCAMIX applied to a set of merely qualitative variables) finds components that discriminate the objects as well as possible in terms of *all* variables, whereas in INDOQUAL each component tends to discriminate the objects mainly in terms of a subset of the variables. For different components different subsets may be involved. As a consequence, INDOQUAL yields components that discriminate the objects better than MCA does, as will be explained below.

Each qualitative variable defines a set of disjoint groups of objects that fall in different categories of the qualitative variable. Hence for each variable one can, for each component, compute the group averages on that component. Now the variance of those group averages is called the "between groups variance", or, because the groups are defined by the categories, the "between categories variance". For both MCA and INDOQUAL it is readily verified that the solution for the matrix of component scores X is centered column-wise. Therefore, for every l, we have $\mathbf{x}_l = J\mathbf{x}_l$. As a consequence, the between categories variance of $\mathbf{x}_l$ with respect to the categories of variable j can be given as

$$\sigma^2_{B(jl)} = n^{-1} \sum_{g=1}^{m_j} f_g(x_l^{gj})^2 = \\ n^{-1}\, \mathbf{x}_l'G_jD_j^{-1}G_j'\mathbf{x}_l = n^{-1}\, \mathbf{x}_l'JG_jD_j^{-1}G_j'J\mathbf{x}_l = c_{jl}, \qquad (30)$$

where f_g denotes the number of objects in category g of variable j, x_l^{gj} denotes the average value of $\mathbf{x}_l$ in category g of variable j, see, for instance, Tenenhaus and Young (1985, p.98). Hence the loading of variable j on component l is equal to the between categories variance for component l, with respect to (the categories defined by) variable j. This between categories variance can also be considered as the amount of discrimination, provided by component l, between the objects that fall in different categories of variable j. Hence the loading c_{jl} indicates how strongly component l discriminates the objects in terms of the categories of variable j.

The above given interpretation of the loadings provides the basis for our statement that the INDOQUAL components discriminate the objects better than the MCA components do. As will be explained below, this difference between INDOQUAL and MCA is an immediate consequence of the fact that INDOQUAL provides a solution with higher simple structure than MCA does, at least in terms of the QMAX, OVERMAX and KURTMAX criteria, and, typically, also in terms of the VMAX criterion.

The results of the previous section on differences between INDOQUAL and MCA in terms of simple structure criteria can be interpreted as follows. The fact that the INDOQUAL loadings have simple structure values that are higher than (or equal to) those of MCA, even after optimal rotation of the MCA solution, implies that INDOQUAL finds loadings that, overall, are more diverse than those resulting from MCA. The loadings are bounded between zero and one. Therefore, a higher simple structure in the loadings implies that more loadings tend to the extreme values, zero and one. Hence INDOQUAL tends to yield more extreme loadings than MCA does. From (30) it also follows that INDOQUAL yields more extreme between categories variances than MCA does. This implies that the INDOQUAL components discriminate the objects better in terms of the categories of *certain* variables (those with large between categories variances), and, at the same time worse in terms of other variables than MCA does. In this way, it can be said that INDOQUAL finds components each of which seek to discriminate the objects, to a larger extent than MCA does, in terms of (possibly different) subsets of variables. Because INDOQUAL seeks to discriminate the objects in terms of fewer variables than MCA does, INDOQUAL will succeed better in actually discriminating the objects. MCA tries to discriminate the objects as well as possible in terms of *all* variables. When certain variables define very different groupings of objects, the MCA components will tend to make compromises by discriminating the objects a little worse both in terms of the one variable and in terms of the other variable. On the other hand, INDOQUAL will optimally discriminate the objects in terms of either one of these "opposite" variables and will possibly discriminate the objects in terms of the other variable by means of a different component.

In addition, it can be said that a subset of variables that load high on a component consists of variables that are, at least in one respect, rather strongly related to each other. That is, if all variables in a subset of variables load high on a component, this component discriminates well the categories of each of these variables. This is only possible if the partitions (groupings) defined by the different variables are highly overlapping. The latter is another way of saying that the qualitative variables involved are highly related with respect to the partition of objects into groups that are best discriminated by the component.

It can be concluded that INDOQUAL finds components that, overall, discriminate the objects better than MCA does, and that it does so by discriminating the objects in terms of the categories of subsets of variables that have highly overlapping partitions.

Above, it has been shown that INDOQUAL finds loadings that tend more to zero and one than those in MCA. However, this does not imply that each component of INDOQUAL *always* has loadings that are greater than those of MCA. Yet, in practice INDOQUAL often yields solutions with loadings that do not only have more simple structure than those of MCA, but that are also, for each component, higher than the highest MCA loadings for a corresponding MCA component. As a consequence, INDOQUAL yields a solution in which, with respect to each component, objects fall apart more clearly than in MCA into (denser) clusters of objects that represent the categories of those highly loading variables.

It can be concluded that INDOQUAL finds loadings that tend more to zero and one than those in MCA. In addition, it has been stated that, in practice, this phenomenon often leads to INDOQUAL components with clusters of objects that are denser and more separated than those (possibly) resulting from MCA. In the next section these phenomena of better "component-wise discriminatory capability" and of "component-wise clearer clustering" are illustrated by means of an example analysis.

8.8. An example analysis of empirical data

The empirical data to be analyzed in the present section has been given by Hartigan (1975, p.228). The data consists of 24 objects like screws and nails, that are classified according to 5 qualitative variables (Whether or not they have a Thread, what type of Head they have, what Indentation they have in the heads, what kind of Bottom they have, and whether or not they are made of Brass). In addition, their Length (in half inches) is measured, which is considered here as a qualitative variable with five categories (1 through 5 half inches). Although the data is of little practical interest, it serves to illustrate the clustering phenomenon, because the objects are well-described in terms of predefined clusters (those of screws, bolts, nails and tacks), whereas this clustering does not refer directly to a qualitative variable in the analysis.

In the present section the MCA solutions and the INDOQUAL solutions for $r = 3$ will be compared. The MCA solution will be considered both before and after varimax rotation. Table 8.1 gives the MCA loadings before varimax rotation and after varimax rotation. It can be seen that the varimax rotation changes the loadings (mainly those of the fifth variable) only slightly, and that these changes lead to increasing simple structure, as expressed by the varimax and quartimax function values. It can also be verified that the amount of explained inertia is equal in the two solutions.

Table 8.1. *MCA loadings before and after varimax rotation.*

	before varimax rotation			after varimax rotation		
	comp.1	comp.2	comp.3	comp.1	comp.2	comp.3
Thread	0.93	0.02	0.00	0.95	0.00	0.00
Head	0.95	0.64	0.74	0.96	0.64	0.73
Head Ind.	0.94	0.67	0.08	0.95	0.74	0.00
Bottom	0.55	0.02	0.00	0.50	0.05	0.01
Length	0.29	0.82	0.69	0.24	0.78	0.79
Brass	0.06	0.03	0.46	0.09	0.00	0.47
MCA-inertia ($\sum_j \sum_l c_{jl}$)		7.90			7.90	
varimax function value		0.34			0.37	
quartimax function value		0.97			1.00	

Table 8.2 gives the loadings for INDOQUAL. Clearly, the components in the INDOQUAL solution and those in the rotated MCA solution have high loadings for the same variables, but those on the INDOQUAL components are higher. This is reflected by the fact that the varimax and quartimax function values are higher for INDOQUAL than for the rotated MCA solution. It can also be seen that the higher simple structure of INDOQUAL is obtained at the cost of a (small) loss in explained inertia compared to the MCA solution.

Table 8. 2. *INDOQUAL loadings.*

	comp.1	comp.2	comp.3
Thread	0.99	0.00	0.00
Head	1.00	0.87	0.81
Head Ind.	0.99	0.31	0.02
Bottom	0.40	0.00	0.02
Length	0.17	0.91	0.82
Brass	0.06	0.00	0.24
MCA–inertia $(\Sigma_j\Sigma_l c_{jl})$		7.61	
varimax function value		0.45	
quartimax function value		1.04	

In both the INDOQUAL and the MCA solutions, the first component is highly correlated with the first three variables. These are the variables that are most important in distinguishing screws and bolts on the one hand from nails and tacks on the other hand. Therefore, the object scores on the first components of both solutions are "plotted" in Figure 8.1. Each of these plots is made in the form of a stem and leaf diagram in which the component scores of the objects are divided into 30 intervals. Because it is of interest to see how well the original clustering in the data appears in the solution, the objects are indicated by the letters T (tack), N (nail), S (screw) and B (bolt).

From inspection of Figure 8.1 it follows that the objects are clustered more clearly with respect to the first INDOQUAL component than with respect to the first MCA component. In addition, the original categories appear as partly separated clusters. That is, the nails and tacks now form one cluster. The bolts and screws form different clusters, but are not separated very much. With respect to the second and third components similar plots could be made, and one would again find a clearer clustering with respect to the INDOQUAL components than with respect to the MCA components. This demonstrates the phenomenon that the INDOQUAL components have a better discriminatory capability than MCA has, as was explained in the previous section.

	MCA	INDOQUAL
0.28 – 0.30		
0.26 – 0.28		
0.24 – 0.26		
0.22 – 0.24	NN	
0.20 – 0.22	NNNNTTN	TNNNNNTTN
0.18 – 0.20		NN
0.16 – 0.18	TN	N
0.14 – 0.16	N	
0.12 – 0.14		
0.10 – 0.12		
0.08 – 0.10		
0.06 – 0.08		
0.04 – 0.06		
0.02 – 0.04		
0.00 – 0.02		
–0.02 – 0.00		
–0.04 – –0.02		
–0.06 – –0.04		
–0.08 – –0.06		
–0.10 – –0.08		
–0.12 – –0.10	SS	
–0.14 – –0.12	S	
–0.16 – –0.14	S	
–0.18 – –0.16	S	SS
–0.20 – –0.18	S	SSSS
–0.22 – –0.20		
–0.24 – –0.22		BBBBBB
–0.26 – –0.24	BBBBB	
–0.28 – –0.26	B	
–0.30 – –0.28		

Figure 8.1. *Stem and leaf diagrams for the object scores on the first components for MCA and INDOQUAL.*

8.9. Discussion

In the present chapter methods have been described for representing mixtures of variables by component scores that optimize simple structure criteria. Two strategies have been proposed. The first one is based on first optimally accounting for the inertia in the variables, and then rotating these component scores (without loss of inertia accounted for) to optimal simple structure. The second strategy consists of finding component scores that primarily optimize simple structure, but will still tend to explain a reasonable amount of inertia. This second strategy has a similar objective as the one that prevails in Projection Pursuit (e.g., Jones & Sibson, 1987), that is, that of revealing clustering in the data. Whether or not the methods proposed here might provide a useful alternative to Projection Pursuit methods is yet to be investigated.

Apart from providing methods to optimize simple structure by means of components for mixed variables, the present chapter describes an important difference between INDOQUAL and MCA. It has been shown that INDOQUAL has a greater discriminatory capability than MCA has. In addition, it has been explained why INDOQUAL tends to yield solutions in which the objects are more clearly clustered, per component, than in MCA solutions. This interpretation of INDOQUAL suggests comparing INDOQUAL with discriminant analysis techniques as well as cluster techniques for qualitative variables. As far as the latter is concerned, comparison with the newly developed GROUPALS technique (Van Buuren & Heiser, 1989) which combines K–means clustering procedures with optimal scaling of qualitative variables is of interest. It should be noted, however, that INDOQUAL has not been developed for purposes of discriminant or cluster analysis, but that the results presented here are merely additional properties of the method that attempts to combine the objectives of optimal representation of the objects (as MCA does) and optimal representation of the qualitative variables (as PCA of quantification matrices does).

In section 8.4 simple structure rotations have been proposed for PCAMIX solutions. Apart from simple structure rotation, an often used rotation technique in ordinary PCA is matching of the loadings to a given set of loadings. A procedure for rotating the MCA object coordinates such that the loadings optimally resemble a given set of loadings has recently been developed, and the program for it is presently being written. This offers new possibilities for comparing INDOQUAL and MCA solutions, because one may now compare the INDOQUAL solution to the rotation of the MCA solution the loadings of which optimally resemble those of the INDOQUAL solution. Furthermore, this matching procedure can be used to choose MCA solutions that optimally resemble an a priori given set of loadings, which is based on theory or on earlier results, for instance.

9. A COMPUTATIONAL SHORT-CUT FOR INDOMIX AND SOME PROPERTIES OF THE INDOMIX SOLUTION

9.1. Introduction

In the chapters 5, 6, 7, and 8 INDOQUAL and INDOMIX have been discussed. Both are based on the application of INDORT to a set of quantification matrices, $P_j = JG_jD_j^{-1}G_j'J$ for qualitative variables, and $Q_j = n^{-1}\mathbf{z}_j\mathbf{z}_j'$ for quantitative variables, respectively, $j = 1,...,m$. In the present chapter it will be described how the solutions of these methods can be obtained. In addition, some particular properties of the solutions will be discussed.

An algorithm for INDORT, that is, for minimizing

$$\sigma(X,W_1,...,W_m) = \sum_{j=1}^{m} \| S_j - XW_jX' \|^2 \tag{1}$$

over X, subject to $X'X = I_r$, has been given by Kroonenberg (1983, p.118). Ten Berge, Knol and Kiers (1988) have suggested an alternative algorithm for which monotone convergence is guaranteed if the matrices S_j are positive semi–definite (p.s.d.). Both algorithms are based on an iterative procedure in which an update for X is computed from the previous X and the set of S_j matrices. When these algorithms are used for INDORT on a set of quantification matrices for qualitative variables one is often faced with severe computational problems. That is, these algorithms are to be applied to a set of m matrices of order $n \times n$. Clearly, computation time increases rapidly as the sizes of the matrices increase. As a consequence, analyzing a set of qualitative variables measured on a *large* number of objects poses immense computational problems, both in terms of required memory and in terms of computation times.

Multiple Correspondence Analysis (MCA) has no such problems in handling very large numbers of objects. This is a consequence of the fact that the information that is essentially used in the computations is contained completely in the Burt–matrix, that is, the supermatrix containing the contingency tables for all pairs of variables, including the diagonal matrices of marginal frequencies. The number of objects in no way affects the size of the Burt-matrix. Hence having a large number of objects in MCA does not give rise to computational problems.

Kiers (1989a) has shown that, as in MCA, the solution of INDOQUAL depends on the elements of the Burt-matrix only. This has been shown by means of describing the algorithm given by Ten Berge et al. (1988) completely in terms of elements of the Burt-matrix. In addition, he has shown that for determining the INDOMIX solution, again, only aggregate information is used during the computations of the solution. In this way, all computational problems for INDOQUAL or INDOMIX due to large sample sizes are resolved. The derivations and results from Kiers (1989a) are repeated here.

The new computational procedure is based on some results that are interesting in itself, because they can be used for deriving some properties of the solutions of INDOQUAL and INDOMIX. For instance, based on a procedure for weighting the objects that is developed here, the "distributional equivalence property" which is a well-known property of Correspondence Analysis, can be shown to hold for INDOQUAL as well. It will also be shown how missing data can be handled, again by applying weights to the objects. First, however, it will be shown that the Ten Berge et al. (1988) algorithm for INDORT applied to p.s.d. quantification matrices uses aggregate information only.

9.2. The Ten Berge, Knol, & Kiers algorithm for INDORT applied to quantification matrices

In the present section, the algorithm proposed by Ten Berge et al. (1988) for INDORT on p.s.d. matrices will be elaborated for the case where this algorithm is applied to quantification matrices S_j. In chapters 5 and 7, particular choices of these quantification matrices have been made. These quantification matrices (as well as most others described in chapter 3) can be decomposed as $S_j = U_jU_j'$. For instance, in the case of the quantification matrices chosen in chapters 5 and 7, $S_j = JG_jD_j^{-1}G_j'J$ for qualitative variables, and $S_j = n^{-1}\mathbf{z}_j\mathbf{z}_j'$ for quantitative variables. Hence U_j can be chosen as $U_j = JG_jD_j^{-½}$ for qualitative variables, and $U_j = n^{-½}\mathbf{z}_j$ for quantitative variables. In the present section, an algorithm will be described for INDORT applied to any set of p.s.d. quantification matrices. In the next section, the implications of the particular choices for the quantification matrices made in chapters 5 and 7 will be studied.

The algorithm proposed by Ten Berge et al. (1988) can be described as follows. Let X^u denote an update of X, then this update is given by

$$X^u = \textstyle\sum_j S_j X W_j \left(\sum_k W_k X' S_k \sum_l S_l X W_l\right)^{-1/2}, \tag{2}$$

where $W_j = \mathrm{Diag} X'S_jX$, for $j = 1,...,m$. Ten Berge et al. (1988) have shown that repeatedly updating X according to (2) monotonically decreases the function σ, or, equivalently, monotonically increases the function

$$\mathrm{f}(X) = \sum_{j=1}^{m} \mathrm{tr}\ (\mathrm{Diag}\ X'S_jX)^2, \tag{3}$$

and because σ is bounded from below (and f from above), this procedure must converge to a stable function value. In (2) the term $\left(\sum_k W_k X' S_k \sum_l S_l X W_l\right)^{-1/2}$ can be computed as $K\Lambda^{-1/2}K'$ from the eigendecomposition $\left(\sum_k W_k X' S_k \sum_l S_l X W_l\right) = K\Lambda K'$.

The present elaboration of the Ten Berge et al. algorithm is based on substituting $S_j = U_jU_j'$ for S_j in the formula for the update X^u. This yields

$$X^u = \textstyle\sum_j U_j U_j' X W_j \left(\sum_k W_k X' U_k U_k' \sum_l U_l U_l' X W_l\right)^{-1/2}. \tag{4}$$

Let $Y_j \equiv U_j'X$, then (4) can be rewritten as

$$X^u = \textstyle\sum_j U_j Y_j W_j \left(\sum_k W_k Y_k' U_k' \sum_l U_l Y_l W_l\right)^{-1/2}. \tag{5}$$

Let $Z_j \equiv Y_jW_j$, $Z \equiv \begin{bmatrix} Z_1 \\ \vdots \\ Z_m \end{bmatrix}$, and $U \equiv (U_1 | ... | U_m)$, then $UZ = \sum_l U_l Z_l$, and (5) can be simplified as

$$\begin{aligned} X^u &= \textstyle\sum_j U_j Z_j \left(\sum_k Z_k' U_k' \sum_l U_l Z_l\right)^{-1/2} \\ &= UZ\left(Z'U'UZ\right)^{-1/2}. \end{aligned} \tag{6}$$

It should be noted that $Z_j \equiv Y_jW_j = Y_j(\mathrm{Diag}\ X'S_jX) = Y_j(\mathrm{Diag}\ X'U_jU_j'X) = Y_j(\mathrm{Diag}\ Y_j'Y_j)$ depends on Y_j only. As a consequence, the update X^u of X depends on the elements of U and Y only.

As has been explained above, the Ten Berge et al. (1988) algorithm for

INDORT is based on iteratively updating X by X^u described in (4). In order to compute the next update X^{uu} of X^u one uses again the expression in (4), but now with every X at the right-hand side replaced by X^u. Expression (4) can again be translated in an expression of the form of (6), when one defines $Y_j^u \equiv U_j'X^u$, $Y^u \equiv \begin{bmatrix} Y_1^u \\ \vdots \\ Y_m^u \end{bmatrix}$, $Z_j^u \equiv Y_j^u(\text{Diag}\, Y_j^{u\prime}Y_j^u)$, and $Z^u \equiv \begin{bmatrix} Z_1^u \\ \vdots \\ Z_m^u \end{bmatrix}$. Then the expression for X^{uu} is given by

$$X^{uu} = UZ^u\left(Z^{u\prime}U'UZ^u\right)^{-1/2}. \tag{7}$$

Clearly, the update X^{uu} for X^u depends on U and Z^u only. The latter itself depends on Y^u only. As a consequence, in order to compute X^{uu} one needs to have Y^u and U only. Obviously, Y^u can be computed from X^u. However, when n is large this computation is very cumbersome. Instead, Y^u is found directly from Z and U, as follows. Premultiplying (6) by U' yields

$$Y^u = U'X^u = U'UZ\left(Z'U'UZ\right)^{-1/2}. \tag{8}$$

From (8) it can be seen that Y^u can be computed by means of Z (which depends on Y only), and U. Matrix Z itself is given by the supermatrix containing $Z_j = Y_j(\text{Diag}\, Y_j'Y_j)$, $j = 1,\ldots,m$. In this expression Y_j depends on U_j and X only. Hence X^{uu} can be computed from X and U without, intermediately, computing X^u.

It follows from the above that the Ten Berge et al. algorithm for updating X can be modified such that it updates X *implicitly* in every iteration, while it only computes an update for Y *explicitly*. The only computations in which X is actually involved are the computation of Y from U and X at the start of the iterative procedure, and the computation of the solution for X from the final update of Y, after convergence of the iterative procedure. During the iterations computations are based on $U'U$ and Z only, while the matrix Z itself depends entirely on Y. Especially when the number of elements of $U'U$ is small compared to mn^2, the present procedure can gain a lot in computation time and memory space needed.

Ten Berge et al. (1988) have proven the monotone convergence of their algorithm by proving that $\text{f}(X^u) \geq \text{f}(X)$, $\text{f}(X^{uu}) \geq \text{f}(X^u)$, etc. In the procedure sketched above, X^u is not explicitly computed. Nevertheless, the corresponding function value can be computed. Substituting $S_j = U_jU_j'$ for S_j in (3) we have

$$\begin{aligned} \mathrm{f}(X^u) &= \sum_{j=1}^{m} \mathrm{tr}\,(\mathrm{Diag}\,X^{u\prime}U_jU_j'X^u)^2 \\ &= \sum_{j=1}^{m} \mathrm{tr}\,(\mathrm{Diag}\;Y_j^{u\prime}Y_j^{u})^2 \equiv \mathrm{g}(Y^u). \end{aligned} \tag{9}$$

Because the algorithm has not been changed essentially by our procedure, it follows from $\mathrm{f}(X^u) \geq \mathrm{f}(X)$, $\mathrm{f}(X^{uu}) \geq \mathrm{f}(X^u)$ that $\mathrm{g}(Y^u) \geq \mathrm{g}(Y)$, $\mathrm{g}(Y^{uu}) \geq \mathrm{g}(Y^u)$, etc. That is, updating Y by means of (8) increases $\mathrm{g}(Y)$ monotonically.

The algorithm proposed here can be summarized as follows.

Initialization:

Step 1. Choose a starting configuration X^0 for X.

Step 2. Compute $Y_j^{0} = U_j'X^0$, $\quad j = 1,\ldots,m$.

Iterations:

Step 3. Compute $Z_j^{i} = Y_j^{i}(\mathrm{Diag}\;Y_j^{i\prime}Y_j^{i})$, $\quad j = 1,\ldots,m$.

Step 4. Compute $Y_j^{i+1} = U_j'UZ^i\left(Z^{i\prime}U'UZ^i\right)^{-1/2}$, $\quad j = 1,\ldots,m$.

Step 5. Evaluate $\mathrm{g}(Y^{i+1}) = \sum_{j=1}^{m} \mathrm{tr}\;(\mathrm{Diag}\;Y_j^{i+1\prime}Y_j^{i+1})^2$.

If $\mathrm{g}(Y^{i+1})\text{-}\mathrm{g}(Y^i) > \varepsilon$, for some small value ε, then go to Step 3, else go to Step 6.

Determine the solution for the final W_j and X:

Step 6. Compute $W_j^{i+1} = (\mathrm{Diag}\;Y_j^{i+1\prime}Y_j^{i+1})$ $\quad j = 1,\ldots,m$.

Step 7. Compute $Z_j^{i+1} = Y_j^{i+1}(\mathrm{Diag}\;Y_j^{i+1\prime}Y_j^{i+1})$, $\quad j = 1,\ldots,m$.

Step 8. Compute $X^{i+2} = UZ^{i+1}\left(Z^{i+1\prime}U'UZ^{i+1}\right)^{-1/2}$.

The bulk of the computations is done in the Steps 3, 4, and 5. These steps only involve (parts of) the matrices $U'U$, of order $\sum_j r_j \times \sum_j r_j$, and Y and Z, both of order $\sum_j r_j \times r$, where r_j is the column-order of U_j. When n is large, the steps in which matrices of row- and/or column-orders n are involved are problematic. Such steps are Steps 1 and 2, and Step 8. In the present algorithm, these steps have to be done only once. In the original Ten Berge et al. algorithm every iteration step involves multiplication of matrices of row– and/or column–orders n. For this reason, the procedure proposed here is much faster than the original procedure when n is large compared to $\sum_j r_j$.

Some details in the algorithm described above need further explanation. The first step of choosing a starting configuration for X can be done on several grounds. Carroll, Pruzansky and De Soete (1987) and more recently, Carroll, De Soete and Pruzansky (1989), have compared the performance of the INDSCAL algorithm when several different starting procedures are used. Their choices for starting matrices for X might be used. However, then one still has to compute Y^0 (Step 2), which might be cumbersome if n is very large. It is easier to omit Step 1, and replace Step 2 by choosing arbitrary matrices Y_j^0, but then a corresponding X^0 does not necessarily exist, and one cannot state that $\mathrm{g}(Y^1) \geq \mathrm{g}(Y^0)$. After the first iteration, however, a column–wise orthonormal X^i that corresponds to Y^i can be defined as $X^i = UZ^{i-1}(Z^{i-1\prime}U'UZ^{i-1})^{-\frac{1}{2}}$. From $\mathrm{f}(X^{i+1}) \geq \mathrm{f}(X^i)$, for $i = 1,2,...$ it follows that $\mathrm{g}(Y^{i+1}) \geq \mathrm{g}(Y^i)$, for $i = 1,2,...$ That is, except for the first step monotone convergence is guaranteed.

In the actual iteration Steps 3 and 4 one uses Y and Z. It should be noted, however, that these need not be stored separately. That is, Y and Z can use the same memory location alternatively, because they are never needed at the same time.

This concludes the description of a modification of the Ten Berge et al. algorithm for INDORT applied to quantification matrices, in general. It can be seen that the algorithm essentially uses the elements of matrix $U'U$ only. In the next section, these sub–matrices will be described for the choices of quantification matrices made in chapters 5 and 7.

9.3. Implications for INDOMIX

In chapters 5 and 7, the quantification matrices have been chosen as $S_j = JG_jD_j^{-1}G_j'J$ for qualitative variables, and $S_j = n^{-1}\mathbf{z}_j\mathbf{z}_j'$ for quantitative variables. Hence U_j can be chosen as $U_j = JG_jD_j^{-\frac{1}{2}}$ for qualitative variables, and as $U_j = n^{-\frac{1}{2}}\mathbf{z}_j$ for quantitative variables. In the present section, the sub–matrices $U_j'U_k$ of $U'U$ will be described for these choices of quantification matrices.

Three cases are to be distinguished. The first case is the case where both variables are qualitative. Then $U_j'U_k$ is given by

$$U_j'U_k = D_j^{-½}G_j'JG_kD_k^{-½}. \tag{10}$$

By substituting $J = (I - n^{-1}11')$ for J, and using $G_j'1 = D_j1$ we can rewrite (10) as

$$\begin{aligned} U_j'U_k &= D_j^{-½}G_j'(I - n^{-1}11')G_kD_k^{-½} \\ &= D_j^{-½}(G_j'G_k - n^{-1}G_j'11'G_k)D_k^{-½} \\ &= D_j^{-½}G_j'G_kD_k^{-½} - n^{-1}D_j^{½}11'D_k^{½}. \end{aligned} \tag{11}$$

The elements of $U_j'U_k$ can be expressed in terms of category frequencies and bivariate category frequencies, as follows. Let f_g be the frequency of category g of variable j, f_h the frequency of category h of variable k, and let f_{gh} be the number of objects that belong to both category g of variable j, and category h of variable k (called the bivariate frequency of these categories). It should be noted that when $j = k$, then $f_{gh} = f_g = f_h$ when $g = h$, and $f_{gh} = 0$, when $g \neq h$. From (11) it follows that the element (g,h) of $U_j'U_k$ is given by

$$[U_j'U_k]_{gh} = f_g^{-½}f_h^{-½}f_{gh} - n^{-1}f_g^{½}f_h^{½}. \tag{12}$$

Clearly, when $j = k$, (12) can be reduced to

$$[U_j'U_k]_{gh} = - n^{-1}f_g^{½}f_h^{½}, \qquad \text{when } g \neq h, \tag{13}$$

and

$$\begin{aligned} [U_j'U_k]_{gg} &= f_g^{-½}f_g^{-½}f_g - n^{-1}f_g^{½}f_g^{½} \\ &= 1 - n^{-1}f_g. \end{aligned} \tag{14}$$

When all variables are qualitative, all sub-matrices of $U'U$ can be computed as in (12), (13), and (14), hence, in that case, one can find the elements of the complete matrix $U'U$ in terms of the category frequencies and the bivariate frequencies. The category frequencies are given on the diagonal of the Burt-matrix (which contains all pairwise contingency tables), and the bivariate frequencies are given in the off-diagonal blocks of the Burt-matrix.

That is, the computations for the solution of INDOQUAL use the elements of the Burt–matrix only. The scores of the objects on the qualitative variables are needed only when the final matrix X is to be computed. However, when the number of objects is large, in general one is not interested in the complete matrix X. A more interesting representation of the results might in that case be the set of category centroids, as proposed by Kiers (1988), and also mentioned in section 5.5. That is, in order to summarize the solution for the objects, one might consider for each variable the means of the object coordinates in each of the categories as given by $D_j^{-1}G_j'X$. Obviously, these category centroids are computed easily from the final values in the matrices $Y_j = D_j^{-½}G_j'JX = D_j^{-½}G_j'X$ after convergence. Hence, if one is satisfied with category centroids only, one does not need to perform Step 8 in the algorithm, which is the only step in which the complete set of scores of objects on variables is needed. An important implication of this is that the present procedure makes possible the analysis of a Burt–matrix while one does not have information on the level of each observation unit. It also follows that the computational efficacy of the algorithm is in no way affected by the number of observation units on which the Burt–matrix is based.

In case some or all variables are quantitative, the sub–matrices of $U_j'U_k$ can be described as follows. If variable j is qualitative and variable k is quantitative, then

$$U_j'U_k = n^{-½}D_j^{-½}G_j'J\mathbf{z}_k = n^{-½}D_j^{-½}G_j'\mathbf{z}_k = n^{-½}D_j^{½}\mathbf{m}_{jk}, \tag{15}$$

where $\mathbf{m}_{jk}$ is the vector with the means of $\mathbf{z}_k$ in each of the categories of variable j.

If variables j and k are both quantitative, then

$$U_j'U_k = n^{-1}\mathbf{z}_j'\mathbf{z}_k = r_{jk}, \tag{16}$$

where r_{jk} is the product–moment correlation between variables j and k.

From the above it follows that the algorithm for INDOMIX involves again aggregate information only, that is, only category frequencies, bivariate frequencies, means of the quantitative variables in the categories of the qualitative variables (given in $\mathbf{m}_{jk}$), and product–moment correlations between quantitative variables. In the case where one has quantitative variables only, the algorithm is based entirely on the product–moment correlations between the variables.

Apart from the fact that the algorithm for INDOMIX (and hence for the special case INDOQUAL) uses aggregate information only, inspection of the algorithm also yields some further information on the INDOMIX solution. That is, from (6), or, equivalently, from Step 8 of the algorithm, it follows that, after convergence of the algorithm, $X = UB$ for some matrix B of order $\sum_j r_j \times r$. That is, the columns of X form an orthonormal basis of a sub–space of the column–space of U. This implies that the maximum number of INDOMIX components is equal to the rank of U. It should be noted, however, that for this maximum number the X matrix does not necessarily provide a perfect INDORT fit. It gives the maximally attainable INDORT–fit in that case. One cannot generally increase the dimensionality of the INDOMIX solution until a perfect INDORT–fit is attained.

In the case of INDOQUAL the rank of U is equal to $\sum_j m_j - m$. Hence the maximal dimensionality for the INDOQUAL solution is $\sum_j m_j - m$. It is well–known that this is the maximal dimensionality for MCA as well. Moreover, as is readily verified, MCA also finds an orthonormal basis for a subspace of the column–space of U. Clearly, when $r = \sum_j m_j - m$, the columns of the INDOQUAL solution for X and those of the MCA solution for X both span the complete column–space of U. It follows that the solution of INDOQUAL for X is equal to the MCA solution for X, up to a possible rotation. Because the MCA solution is determined up to a rotation only, the INDOQUAL solution for $r = \sum_j m_j - m$ is also an MCA solution. It is readily verified that, when the MCA solution would be rotated such that it maximizes the quartimax criterion (see section 8.4) it would yield the INDOQUAL solution itself.

In the case of mixed variables a similar equivalence can be shown to hold for INDOMIX and PCAMIX. A more interesting case, however, seems to be the one where only quantitative variables are involved. In that case $U = n^{-1/2}Z$, hence $X = ZB$ for some $m \times r$ matrix B. That is, the components resulting from INDOMIX applied to quantitative variables, given in X, are linear combinations of Z, just as in ordinary PCA. Therefore, ordinary PCA and INDOMIX applied to quantitative variables can be seen as methods that both find linear combinations of the variables, but by optimizing different criteria. From the fact that the components from INDOMIX are linear combinations of the variables it follows that one can compute component scores for new (or "supplementary") objects as well. This creates the possibility of cross–validation of one's results by applying the component–weights (in matrix B) resulting from an INDOMIX analysis in one sample to the variables of another sample, in order to study how the components are recovered in this second sample. Such a cross–validation can be useful for several purposes. For instance, one can use cross–validation to determine how sensitive an INDOMIX solution is to the particular sample on which the solution is based, as is done in section 10.4. Alternatively, one may want to study how strongly components that have been determined at one occasion are recovered at another occasion.

9.4. A further simplified algorithm for INDOQUAL

In section 5.4, it has been mentioned that INDOQUAL can be seen as the method that applies INDORT to the matrices $P_j = G_jD_j^{-1}G_j'$, $j = 1,...,m$, and eliminates the trivial axis, which is found consistently in this INDORT analysis. In the present section, a computational procedure for INDORT applied to the matrices $P_j = G_jD_j^{-1}G_j'$ will be given, that appears to be a little more simple than the one described for INDOQUAL above.

The INDORT analysis of the P_j matrices could be performed along similar lines as that of the S_j matrices. That is, one sets $U_j = G_jD_j^{-½}$. Again the algorithm from the previous section is used and the elements of the blocks of $U'U$ are computed as

$$[U_j'U_k]_{gh} = f_g^{-½}f_h^{-½}f_{gh}. \qquad (17)$$

Clearly, when $j = k$, $U_j'U_k$ can be written as

$$U_j'U_j = D_j^{-½}G_j'G_jD_j^{-½} = I_{m_j}. \qquad (18)$$

The algorithm of the previous section depends on the $U'U$ matrix whose elements are given as in (17) and (18). These elements are the elements of the Burt–matrix divided by the square roots of their associated marginal frequencies. It would be even simpler when the algorithm used the elements of the Burt–matrix itself. Therefore, in practice we use an algorithm which is a slight modification of the algorithm sketched above. Let $\tilde{U}_j \equiv G_j = U_jD_j^{½}$,

$\tilde{Y}_j \equiv D_j^{-1}G_j'X = D_j^{-½}Y_j$, and $\tilde{Z}_j \equiv \tilde{Y}_j(\text{Diag } \tilde{Y}_j'D_j\tilde{Y}_j) = D_j^{-½}Y_j(\text{Diag } Y_j'Y_j) = D_j^{-½}Z_j$. Then $\tilde{U}_j\tilde{Z}_j = U_jZ_j$, and $\tilde{U}\tilde{Z} = UZ$, with $\tilde{U}$ the horizontal supermatrix of the $\tilde{U}_j$ matrices, and $\tilde{Z}$ the vertical supermatrix of the $\tilde{Z}_j$ matrices. Then the iteration Steps 3, 4 and 5 are to be replaced by

$$\text{Step 3': Compute } \tilde{Z}_j^{\,i} = D_j^{-½}Z_j^{\,i} = D_j^{-½}Y_j^{\,i}(\text{Diag } Y_j^{\,i\prime}Y_j^{\,i})$$

$$= \tilde{Y}_j^{\,i}(\text{Diag } \tilde{Y}_j^{\,i\prime}D_j\tilde{Y}_j^{\,i}), \qquad j = 1,\ldots,m \ ;$$

$$\text{Step 4': Compute } \tilde{Y}_j^{\,i+1} = D_j^{-½}Y_j^{\,i+1} = D_j^{-½}U_j'UZ^i\left(Z^{i\prime}U'UZ^i\right)^{-½}$$

$$= D_j^{-1}\tilde{U}_j'\tilde{U}\tilde{Z}^i\left(\tilde{Z}^{i\prime}\tilde{U}'\tilde{U}\tilde{Z}^i\right)^{-½}, \qquad j = 1,\ldots,m \ ;$$

$$\text{Step 5': Evaluate } g'(\tilde{Y}^{i+1}) = \sum_{j=1}^{m} \text{tr } (\text{Diag } \tilde{Y}_j^{\,i+1\prime}D_j\tilde{Y}_j^{\,i+1})^2.$$

Steps 2, 6, 7, and 8 are to be adapted analogously. The advantage of using the present procedure over the original one is that in Step 4′ one uses $\tilde{U}'\tilde{U}$ only which is exactly the Burt–matrix. It should be noted that the matrices D_j are the block–diagonal matrices of the Burt–matrix, that is, $D_j = \tilde{U}_j'\tilde{U}_j$. In this way it suffices to work with the elements of the Burt–matrix $\tilde{U}'\tilde{U}$ which are integers, instead of the elements of $U'U$, which are reals and hence require more memory space. This might enhance computer–efficiency, although the advantage is off–set by the disadvantage of a more complicated computation of the $\tilde{Z}_j$ matrices, and of the function to be evaluated. Incidentally, it should be noted that the matrix $\tilde{Y}_j$ is the matrix with category centroids. This matrix is now computed during the iterations, and it is available at once after convergence.

9.5. Applying weights to the objects by requiring distributional equivalence

In the previous sections it has been shown that the solution of INDOQUAL is based entirely on the elements of the Burt–matrix. Obviously, two objects with identical scores on all variables contribute in exactly the same way to the Burt–matrix. Hence two such objects can be seen as one "type of object" that occurs twice. Another way of putting this is that the object occurs with weight 2 in the sample. In general, one can consider p objects that have the same scores on all variables as one object with weight p. The elements of the Burt–matrix are then computed as follows. Let object i have weight p_i, then the element of $[B_{jk}]_{gh}$, that is the bivariate frequency of category g of variable j, and category h of variable k, is given by $[B_{jk}]_{gh} = \sum_{(i \in GH)} p_i$, that is, the sum of weights of the objects that fall in category g of variable j, and category h of variable k. Clearly, it makes no difference whether all objects with the same scores on all variables are mentioned explicitly, with unit weights, or all p objects with equal scores are replaced by *one* object with weight p. This property is similar to the well–known property of "distributional equivalence" in Correspondence Analysis (CA). That is, when two rows (or columns) are proportional, CA on the matrix in which these rows (or columns) are replaced by their sum yields the same solution (e.g., Greenacre, 1984, p.95). This in turn comes down to CA on the same data with the two equal rows (or columns) replaced by one with a weight of 2. This property can readily be generalized to MCA, which, just as INDOQUAL, depends on the Burt–matrix only.

The property of distributional equivalence can be useful for INDOQUAL when a (small) number of profiles is given that are each observed many times in different frequencies. Because the program uses the Burt–matrix only and allows for differential weighting of the objects, such data can be analyzed without problem. In principle, the property of distributional equivalence also holds for INDOMIX, but it is rather unlikely that two objects have identical scores on quantitative variables. Using weights for the objects, however, opens an interesting possibility for modifying INDOQUAL and INDOMIX. That is, instead of using the weights only to summarize a number of equal profiles, one can make any choice of weights for the objects, for instance in order to down–weight the influence of certain objects that should not affect the solution unduly, or, conversely, to give them a high weight in order to let them dominate the solution to a certain extent. Another interesting application seems to be a kind of "fuzzy coding". That is, an object can be seen to belong to more than one category of a variable, to different extents. Such a fuzzy coding can be applied in INDOQUAL by replacing each object by a number of "sub–objects" (with weights summing to one) which all belong to different categories of the variable which is to be coded fuzzily (cf. Cazes, 1980, pp.391–392). The weights of the sub–objects indicate the extent to which the object belongs to each of the categories. One particular application of such a fuzzy coding can be used in the case of missing data, as is explained in the next section. It should be noted that this type of fuzzy coding differs from the more usual definition of fuzzy coding (e.g., Van Rijckevorsel, 1987, p.104 ff). Usually, fuzzy coding comes down to replacing indicator matrices by so-called "pseudo-indicator matrices", which contain in each row a number of weights summing to one, indicating to what extent an object belongs to the category. Obviously, this second type of fuzzy coding can be incorporated in INDOMIX as well, because it simply pertains to a particular choice of quantification matrices, based on pseudo-indicator matrices.

9.6. Missing data

In the descriptions of INDOQUAL and INDOMIX no procedure has been described for handling missing data. Of course, one simple way of handling missing data is deletion of all objects for which scores on one or more of the variables are missing, often called "list-wise deletion". However, in case the number of variables is large, even with a small percentage of missing data, this procedure of "list-wise deletion" might come down to eliminating most (or even all) data. Therefore, alternative strategies are desirable.

For INDOQUAL, one particularly simple method for handling missing data is based on the procedure for fuzzy coding described in section 9.5. Suppose an object's score is missing for variable j only. Then, not knowing to which category this object belongs, one might consider this object to belong (to a certain extent) to all the categories of the variable. That is, each object is replaced by m_j "sub-objects" with certain weights adding up to 1, that belong each to a different category of variable j. The choice of weights to assign to the sub-objects might be inspired by various reasonings. A very simple procedure is to assign the weight m_j^{-1} to all sub-objects. However, if a category is rather infrequent, it is reasonable to assign a smaller weight to the sub-object falling in this category. This might be achieved by assigning the weight f_g/n to the sub-object that falls in category g, $g = 1,...,m_j$. Alternatively, one might have some information on the reason why an observation is missing. For instance, an observation on an object might be "missing" because the object in fact belongs partly to category g and partly to category h of a variable, but certainly not to any of the other categories.

Then introducing only two sub–objects (falling in categories g and h, respectively) seems to be called for, for instance with weights of 1/2 each. In all cases, the object scores for objects with missing data can be computed as the means of the object scores for the sub–objects belonging to the object concerned.

In INDOMIX the procedures described above can be used for handling missing observations on qualitative variables. In order to use such procedures for quantitative variables as well one is faced with the problem that, in principle, the number of available scores is infinite. In order yet to align the procedure for handling missing observations on quantitative variables as much as possible to the one for handling missing observations on qualitative variables, one might replace an object with a missing observation on a quantitative variable by as many sub–objects as there are scores that have been observed, with weights summing to one. It is readily verified that, when the weights are proportional to those of the observed scores on this variable, then the values of $\mathbf{z}_j'\mathbf{z}_k$ and $\mathbf{m}_{jk}$ are the same as those obtained by setting the missing observation to this variable equal to zero. Hence one can simply replace missing observations on quantitative variables by scores zero. Another way of interpreting this is by considering a missing observation to be replaced by the mean score on the variable concerned. This procedure in fact unstandardizes the variable, and it seems useful to standardize the variable again after this procedure.

Alternative procedures for handling missing data are possible as well (see Gifi, 1981, pp. 68–70, and, Meulman, 1982, for MCA). One of these is to replace the row in the indicator matrix for a missing observation on a qualitative variable by a row with zero elements only. Still another approach is to create one or more extra categories for missing observations on qualitative variables. Depending on the reason why an observation is missing one may choose for any of these options. None of them involves considerable adaptations of the computational procedures for INDOQUAL or INDOMIX.

9.7. Discussion

In the present chapter an algorithm has been described for the INDORT analysis of a set of p.s.d. quantification matrices. The algorithm described here is based on the algorithm proposed by Ten Berge et al. (1988). One of the main problems with their algorithm, and hence with the one described here, is the fact that it is not guaranteed that the algorithm finds the global maximum of function f. Apart from using rational starts, as proposed by Carroll, Pruzansky, and De Soete (1987) and Carroll, De Soete, and Pruzansky (1989), the only (partial) remedy to this problem seems to be using several restarts.

The algorithm for INDORT applied to quantification matrices has been elaborated for the special cases of INDOQUAL and INDOMIX, but can clearly be useful for INDORT applied to many of the other quantification matrices discussed in chapter 3 as well. Furthermore, as has been discussed in chapter 4, INDORT is not the only three–way method that has been proposed to use for the analysis of a set of quantification matrices for qualitative variables. Marchetti (1988) has proposed to use Tucker's three–mode scaling (Tucker, 1972, see Kroonenberg, 1983, pp.52–53) and IDIOSCAL (Carroll & Chang, 1972) for the analysis of a set of quantification matrices. In addition, one might apply INDSCAL, that is, the unconstrained variant of INDORT, to such a set of quantification matrices. All these methods use algorithms that have computational problems when faced with m large $n \times n$ matrices. Currently, algorithms are being developed for these methods that, just as the algorithm described in the present chapter, need category frequencies and bivariate frequencies only.

An interesting implication of the fact that the INDOQUAL solution is based on the elements of the Burt–matrix only (that is, on category frequencies and bivariate frequencies) is that different data sets with the same Burt–matrix have essentially the same solution. That is, these solutions have the same W_j matrices, and the same category centroids, given by $D_j^{-1}G_j'X$. The only difference is to be found in the object coordinates. This reflects the situation in ordinary PCA, where the PCA of two sets of variables with the same correlation matrix yield the same loadings for the variables, although the component scores may differ.