PSYCHOMETRIKA—VOL. 54, NO. 3, 467–473
SEPTEMBER 1989

ALTERNATING LEAST SQUARES ALGORITHMS FOR SIMULTANEOUS COMPONENTS ANALYSIS WITH EQUAL COMPONENT WEIGHT MATRICES IN TWO OR MORE POPULATIONS

HENK A. L. KIERS
JOS M. F. TEN BERGE

UNIVERSITY OF GRONINGEN

Millsap and Meredith (1988) have developed a generalization of principal components analysis for the simultaneous analysis of a number of variables observed in several populations or on several occasions. The algorithm they provide has some disadvantages. The present paper offers two alternating least squares algorithms for their method, suitable for small and large data sets, respectively. Lower and upper bounds are given for the loss function to be minimized in the Millsap and Meredith method. These can serve to indicate whether or not a global optimum for the simultaneous components analysis problem has been attained.

Key words: simultaneous components analysis, alternating least squares, principal components analysis, longitudinal data, cross-sectional data, stationary component weights.

Principal components analysis (PCA) is a technique for constructing a number of components from a set of variables such that components are found that represent the original variables as well as possible (Meredith & Millsap, 1985). In ordinary PCA components are determined for variables that have been observed in a single population. When observations have been made in more than one population, then it is possible, of course, to analyze the observations separately for each of the populations. However, this will generally yield components for the different populations that are constructed in different ways. That is, components in each population are computed as weighted sums of the variable scores with different weights in each population. As a consequence, the components found across populations need not have much in common.

In order to find components that do have much in common, components can be constructed that are based on the same set of weights for the variables in all populations. As in ordinary PCA, these components can be constructed such that they explain as much variance as possible in all populations simultaneously. We will denote the latter analysis by the heading "simultaneous components analysis" (SCA). For details on the rationale behind SCA and suggestions for interpreting results, as well as for an application of the method to empirical data we refer to Millsap and Meredith (1988).

Millsap and Meredith (1988) have not only proposed a model for simultaneous components analysis. They also have offered an algorithm for fitting this model. However, there is a problem in the algorithm they use to compute the components for SCA. Their algorithm for SCA is based on a conjugate gradient procedure which requires certain identification constraints on the weights. These constraints might be chosen such that the optimal weights can never be found by the procedure (Millsap & Meredith). Moreover, their algorithm has not been shown to converge monotonely.

Financial support by the Netherlands organization for scientific research (NWO) is gratefully acknowledged.

Requests for reprints should be sent to Henk A. L. Kiers, Department of Psychology, Grote Markt 31/32, 9712 HV Groningen, THE NETHERLANDS.

0033-3123/89/0900-9523$00.75/0

The chief objective of the present paper is to describe two alternating least squares (ALS) algorithms for SCA. ALS algorithms converge monotonely by definition. The algorithms developed here require no identification constraints whatsoever. Moreover, lower and upper bounds are formulated for the loss function that is minimized in SCA. The upper bound is used for obtaining a rational start for the parameters that are computed iteratively. The lower bound can be used for providing an indication of the quality of the solution.

An Alternating Least Squares Algorithm for SCA

The SCA problem is to minimize the sum of the residual (or unexplained) variances over all populations. Let X_i denote the n_i by m data matrix from population i, for n_i objects having scores on m variables that are the same across k populations. Let matrix B denote the m by p component weight matrix for all k populations, and let P_i denote the m by p pattern matrix for population i. The pattern matrix gives the least squares weights to be assigned to the components for optimally reconstructing the original variables. Then X_iBP_i' denotes the part of the data matrix X_i that is explained by the components (columns of X_iB) in population i. Therefore, in order to minimize the sum of the residual variances over all populations, the function

$$f(B, P_1, \ldots, P_k) = \sum_{i=1}^{k} \|X_i - X_iBP_i'\|^2 \tag{1}$$

has to be minimized over matrix B and the matrices $P_1, \ldots, P_k$. It will now be shown that an ALS algorithm can be developed in which first the matrices $P_1, \ldots, P_k$ are updated while matrix B is fixed, then matrix B is updated while the matrices $P_1, \ldots, P_k$ are fixed. This procedure is repeated until the function value does not decrease considerably anymore.

For $i = 1, \ldots, k$, minimizing function f over matrix P_i while matrix B is fixed is achieved by solving a multiple regression problem for P_i. It follows from linear regression theory that, assuming that $B'X_i'X_iB$ is nonsingular, the update for P_i is $X_i'X_iB(B'X_i'X_iB)^{-1}$. In case $B'X_i'X_iB$ is singular, a generalized inverse should be used instead of the inverse. This does not essentially change the algorithm and theory developed here.

The problem of minimizing f over B while the matrices P_i are fixed can be solved as follows. Function f consists of a sum of squared euclidean norms of residual matrices. Obviously, these norms do not change when the matrices are strung out as column-vectors containing the elements of the successive rows of the matrices. Therefore, function f can be written as

$$f(B, P_1, \ldots, P_k) = \sum_{i=1}^{k} \|\text{Vec } X_i - \text{Vec } (X_iBP_i')\|^2, \tag{2}$$

where Vec ( ) denotes a matrix strung out row-wise into a column-vector. From elementary algebra it follows that Vec $(X_iB_i'P_i') = (X_i \otimes P_i)$ Vec B, where $\otimes$ denotes the Kronecker product. Using this notation and putting the column-vectors for each of the k populations into one supervector yields

$$f(B, P_1, \ldots, P_k) = \left\| \begin{pmatrix} \text{Vec } X_1 \\ \vdots \\ \vdots \\ \text{Vec } X_k \end{pmatrix} - \begin{pmatrix} X_1 \otimes P_1 \\ \vdots \\ \vdots \\ X_k \otimes P_k \end{pmatrix} \text{Vec } B \right\|^2 . \tag{3}$$

It is now obvious that the problem of minimizing function f over B while the matrices $P_1, \ldots, P_k$ are fixed is a multiple regression problem, with Vec B containing the regression weights. Therefore, function f is minimized over B by choosing

$$\text{Vec } B = \left\{ \sum_{i=1}^{k} (X_i \otimes P_i)'(X_i \otimes P_i) \right\}^{-1} \left\{ \sum_{i=1}^{k} (X_i \otimes P_i)' \text{ Vec } X_i \right\}$$

$$= \left(\sum_{i=1}^{k} X_i'X_i \otimes P_i'P_i \right)^{-1} \text{Vec} \left(\sum_{i=1}^{k} X_i'X_iP_i \right). \tag{4}$$

Subsequently, the update of matrix B is found by simply rewriting Vec B in matrix form.

We have now described a solution to the problem of minimizing function f over matrix P_i while B is fixed, for $i = 1, \ldots, k$, and the problem of minimizing f over B while the matrices $P_1, \ldots, P_k$ are fixed. These $k + 1$ problems are all solved in the least squares sense. Therefore, alternating these procedures yields an alternating least squares algorithm for minimizing function f. This algorithm decreases function f monotonely, and because the function is bounded from below (by zero) the function must converge.

This algorithm requires determining the inverse of a matrix of order mp by mp. For small m and p there is no problem. However, if the number of variables and the number of components required increases, computational efficiency rapidly decreases, due to the necessity of inverting an increasingly large matrix. For this reason, an alternative algorithm is proposed, that requires the inverse of matrices of smaller order (m by m).

Alternating Least Squares Algorithm for SCA on Large Numbers of Variables

In order to handle cases where mp is large, that is, requiring too much computer time or storage, an algorithm has been developed that uses a different alternating least squares procedure. In this algorithm the matrices $P_1, \ldots, P_k$ are updated as in the previous algorithm, but matrix B is updated column-wise. That is, each column of B is updated successively, while the other columns are fixed. It should be noted that the solution for matrix B found during the process is not the best least squares solution for B. However, because all columns of B are optimal in the least squares sense, the function f is decreased nonetheless. This results in an alternating least squares algorithm consisting of $k + p$ steps (p steps for matrix B instead of 1).

The column-wise procedure for updating matrix B will be explained after rewriting function $f(B, P_1, \ldots, P_k)$ in order to isolate the columns of B. Let $\mathbf{b}_h$ denote column h of matrix B, and $\mathbf{p}_{ih}$ column h of matrix P_i, for $h = 1, \ldots, p$. Then $f(B, P_1, \ldots, P_k)$ can be rewritten as

$$f(B, P_1, \ldots, P_k) = \sum_{i=1}^{k} \left\| X_i - \sum_{h=1}^{p} X_i \mathbf{b}_h \mathbf{p}_{ih}' \right\|^2 . \tag{5}$$

Write $f(\mathbf{b}_j)$ to denote that function $f(B, P_1, \ldots, P_k)$ is to be minimized over column $\mathbf{b}_j$ only, while the other columns of B, and the matrices $P_1, \ldots, P_k$ are fixed. Isolating the part containing $\mathbf{b}_j$ in (5) yields

$$f(\mathbf{b}_j) = \sum_{i=1}^{k} \left\| (X_i - \sum_{h \neq j} X_i \mathbf{b}_h \mathbf{p}'_{ih}) - X_i \mathbf{b}_j \mathbf{p}'_{ij} \right\|^2. \tag{6}$$

We define $X_{i(-j)} = (X_i - \Sigma_{h \neq j} X_i \mathbf{b}_h \mathbf{p}'_{ih})$ and simplify (6) as

$$\begin{aligned} f(\mathbf{b}_j) &= \sum_{i=1}^{k} \|X_{i(-j)} - X_i \mathbf{b}_j \mathbf{p}'_{ij}\|^2 \\ &= \left\| \begin{pmatrix} \text{Vec } X_{1(-j)} \\ \vdots \\ \vdots \\ \text{Vec } X_{k(-j)} \end{pmatrix} - \begin{pmatrix} X_1 \otimes \mathbf{p}_{1j} \\ \vdots \\ \vdots \\ X_k \otimes \mathbf{p}_{kj} \end{pmatrix} \mathbf{b}_j \right\|^2. \end{aligned} \tag{7}$$

Minimizing expression (7) over $\mathbf{b}_j$ is a simple linear regression problem. Clearly, the update for $\mathbf{b}_j$ is

$$\begin{aligned} \mathbf{b}_j &= \left[\sum_{i=1}^{k} (X_i \otimes \mathbf{p}_{ij})'(X_i \otimes \mathbf{p}_{ij}) \right]^{-1} \left[\sum_{i=1}^{k} (X_i \otimes \mathbf{p}_{ij})' \text{ Vec } X_{i(-j)} \right] \\ &= \left[\sum_{i=1}^{k} (X'_i X_i) \otimes (\mathbf{p}'_{ij} \mathbf{p}_{ij}) \right]^{-1} \left[\sum_{i=1}^{k} \text{Vec } (X'_i X_{i(-j)} \mathbf{p}_{ij}) \right] \\ &= \left[\sum_{i=1}^{k} \mathbf{p}'_{ij} \mathbf{p}_{ij} X'_i X_i \right]^{-1} \sum_{i=1}^{k} X'_i X_{i(-j)} \mathbf{p}_{ij}. \end{aligned} \tag{8}$$

It should be noted that this algorithm has the advantage that it does not use matrices of order mp by mp. The largest matrix that has to be inverted in this algorithm is an m by m matrix, which allows handling large numbers of variables.

Lower and Upper Bounds to the SCA Loss Function

In order to find lower and upper bounds to loss function (1), we reformulate (1). It is important to note that minimizing the least squares function (1) is equivalent to maximizing the trace of a matrix. That is, for $i = 1, \ldots, k$, P_i can be expressed uniquely in terms of X_i and B as

$$P_i = X'_i X_i B (B' X'_i X_i B)^{-1}, \tag{9}$$

without loss of optimality. Therefore, minimizing (1) over B and $P_1, \ldots P_k$ reduces to maximizing

$$g(B) \equiv \text{tr} \sum_{i=1}^{k} B' C_i^2 B (B' C_i B)^{-1} \tag{10}$$

over B, where we have written C_i for $X_i'X_i$. In fact, it is this form in which Millsap and Meredith (1988) have dealt with the SCA problem (unweighted case).

An upper bound to the minimum of f is obtained as follows. According to ten Berge (1986, p. 56; Lemma 2), we have for every positive definite matrix C_i

$$\text{tr } B'C_i^2B(B'C_iB)^{-1} \geq \text{tr } B'C_iB(B'B)^{-1}, \tag{11}$$

assuming that the inverses exist. Summing over i yields

$$g(B) = \sum_{i=1}^{k} \text{tr } B'C_i^2B(B'C_iB)^{-1} \geq \text{ tr } B'CB(B'B)^{-1}, \tag{12}$$

where C is defined as $C \equiv \Sigma_{i=1}^k C_i$. It readily follows from (12) that the maximum of g is larger than or equal to the maximum of tr $B'CB(B'B)^{-1}$. This trace is maximal when B contains the first p eigenvectors of C, and the maximum is equal to the sum of the first p eigenvalues of C. Therefore, the sum of the first p eigenvalues of C is a lower bound to $g(B)$. Because $f(B)$ = tr $C - g(B)$, the sum of the *last* $m - p$ eigenvalues of C is an *upper* bound to the minimum of f, where $f(B)$ denotes $f(B, P_1, \ldots, P_k)$ with (9) substituted for P_i, $i = 1, \ldots k$.

The upper bound to the minimum of f derived above will be used to choose starting values for the weight matrix B. A matrix B for which this upper bound is attained is given by the matrix containing the first p eigenvectors of C. Choosing this matrix as a starting matrix for B limits the range of possible function values to be passed by the algorithms and decreases the liability to convergence to a local minimum.

A lower bound to function f is readily obtained as follows. Obviously,

$$\text{tr } B'C_i^2B(B'C_iB)^{-1} \leq \sum_{j=1}^{p} \mu_j(C_i),$$

where $\mu_j(C_i)$ is the j-th eigenvalue of C_i, and

$$\text{tr } C_i = \sum_{j=1}^{m} \mu_j(C_i).$$

Hence

$$[\text{tr } C_i - \text{tr } B'C_i^2B(B'C_iB)^{-1}] \geq \sum_{j=p+1}^{m} \mu_j(C_i).$$

From this inequality it follows immediately that

$$f(B) \geq \sum_{i=1}^{k} \sum_{j=p+1}^{m} \mu_j(C_i). \tag{13}$$

Therefore, the right hand side of (13) is a lower bound to the loss function that is to be minimized in SCA.

Comparing the lower bound expressed by (13) to the function value that one obtains by SCA in fact gives the difference between unexplained variance by SCA and unexplained variance by separate PCA analyses in all populations. This difference expresses to what extent the separate PCA's explain what is specific in each of the populations. In order to see how large the difference is in each population it is instructive to compute the amount of variance explained by SCA in each population, and to compare this amount with the amount of variance explained by separate PCA's in each population.

Orthogonality and Rotation of the SCA Components

In ordinary PCA the principal components are mutually orthogonal. The components resulting from SCA, on the other hand, are not orthogonal in any of the populations. Obviously, a nonsingular transformation of the weight matrix can make the components orthogonal in one population, but generally not in all populations at the same time. Such a nonsingular transformation does not change the residuals and function value, when the matrices $P_1, \ldots, P_k$ are subjected to the inverse transformation. In order to prevent an asymmetric treatment of the populations one may choose the components orthogonal over the union of the k populations. That is, the weight matrix might be transformed such that component scores computed for all individuals (from all populations) are orthogonal.

Performance and Application of the ALS Algorithms for SCA

Both SCA algorithms based on ALS have been programmed on a CDC Cyber. When submitted to a series of 14 test data sets (of order 6 × 6), the two ALS algorithms reached the same function value, throughout. In all cases the column-wise algorithm needed considerably less computation time (mean computation time 1.58 sec.) than the global ALS algorithm that updates matrix B at once (mean computation time 5.23 sec). So apart from space limitations for the global algorithm, computation times seem to be another reason to prefer the column-wise algorithm.

As an example we reanalyzed the data set that Millsap and Meredith (1988) used to illustrate their method. This data set consists of two samples of subjects tested on three occasions, thus yielding six independently measured data sets. On each occasion scores on six subtests have been computed for each subject.

This data has been subjected to our SCA procedure. The weight matrix that has been found could be transformed by means of a nonsingular transformation in such a way that the weights reported by Millsap and Meredith (1988) were reproduced. As has been mentioned above, such a nonsingular transformation does not affect the function value. Therefore, our analysis has shown that the solution given by Millsap and Meredith (1988) for their data can just as well be obtained by an algorithm free of identification constraints.

Discussion

It has been explained above that the weight matrix cannot be determined uniquely. That is, any nonsingular transformation of the weight matrix yields the same residuals and function value when the pattern matrices are subjected to the inverse transformation. Which transformation of the weight matrix is to be preferred remains yet to be investigated.

In contrast to the solutions of ordinary PCA, the solutions of SCA for different numbers of dimensions are not nested. That is, the solution obtained for a certain number of components does not necessarily comprise all solutions of an analysis in which a smaller number of components is used. As a consequence, determining SCA solutions with different dimensionalities requires separate computations for each of the dimensionalities.

In the present paper we have given some information on the performance of the two ALS algorithms provided here. On the basis of this information one might conjecture that the column-wise algorithm is not only more useful for handling large data sets, but that the column-wise algorithm is to be preferred to the global algorithm for analyzing data sets of any conceivable size.

For both ALS algorithms presented here convergence to a stationary point is guaranteed. However, it cannot be guaranteed that the global minimum will be attained. Therefore, it is suggested to run more than one analysis on the same data set with different starting configurations for the weight matrix.

References

Meredith, W., & Millsap, R. E. (1985). On component analysis. *Psychometrika*, *50*, 495–507.

Millsap, R. E., & Meredith, W. (1988). Component analysis in cross-sectional and longitudinal data. *Psychometrika*, *53*, 123–134.

ten Berge, J. M. F. (1986). Rotation to perfect congruence and the cross-validation of component weights across populations. *Multivariate Behavioral Research*, *21*, 41–64; 262–266.

Manuscript received 9/21/87
Final version received 7/8/88