JOURNAL OF CHEMOMETRICS, VOL. 1, 41–56 (1987)

MULTI-WAY PRINCIPAL COMPONENTS- AND PLS-ANALYSIS*

SVANTE WOLD
Research Group for Chemometrics, Umeå University, Sweden

PAUL GELADI AND KIM ESBENSEN
Research Group for Chemometrics, Umeå University, Sweden and Norsk Regnesentral, Blindern, Oslo 3, Norway

AND

JERKER ÖHMAN
Research Group for Chemometrics, Umeå University, Sweden

SUMMARY

The Lohmöller–Wold decomposition of multi-way (three-way, four-way, etc.) data arrays is combined with the non-linear partial least squares (NIPALS) algorithms to provide multi-way solutions of principal components analysis (PCA) and partial least squares modelling in latent variables (PLS).

The decomposition of a multi-way array is developed as the product of a score vector and a loading array, where the score vectors have the same properties as those of ordinary two-way PCA and PLS. In image analysis, the array would instead be decomposed as the product of a loading vector and an image score matrix.

The resulting methods are equivalent to the method of unfolding a multi-way array to a two-way matrix followed by ordinary PCA or PLS analysis. This automatically proves the eigenvector and least squares properties of the multi-way PCA and PLS methods.

The methodology is presented; the algorithms are outlined and illustrated with a small chemical example.

KEY WORDS Multi-way array Multiorder array Principal components PLS Multivariate calibration

INTRODUCTION

In multivariate data analysis, principal components analysis (PCA) of a data matrix **X** is a basic tool. PCA decomposes **X** into a score matrix **T** times a loading matrix **P** plus a residual matrix **E** (Figure 1):

$$\mathbf{X} = \mathbf{T}\mathbf{P}' + \mathbf{E} \tag{1}$$

This decomposition is particularly useful for converting **X** to a few informative plots (score plots and loading plots) and for modelling the systematic structure in **X**. PCA is equivalent to singular value decomposition (SVD) and closely related to factor analysis.[1–5]

*A preliminary version of this paper was presented at the Nordic Symposium for Applied Statistics, RECKU, Copenhagen, January 1986.

0886–9383/87/010041–16$08.00 *Received 8 July 1986*

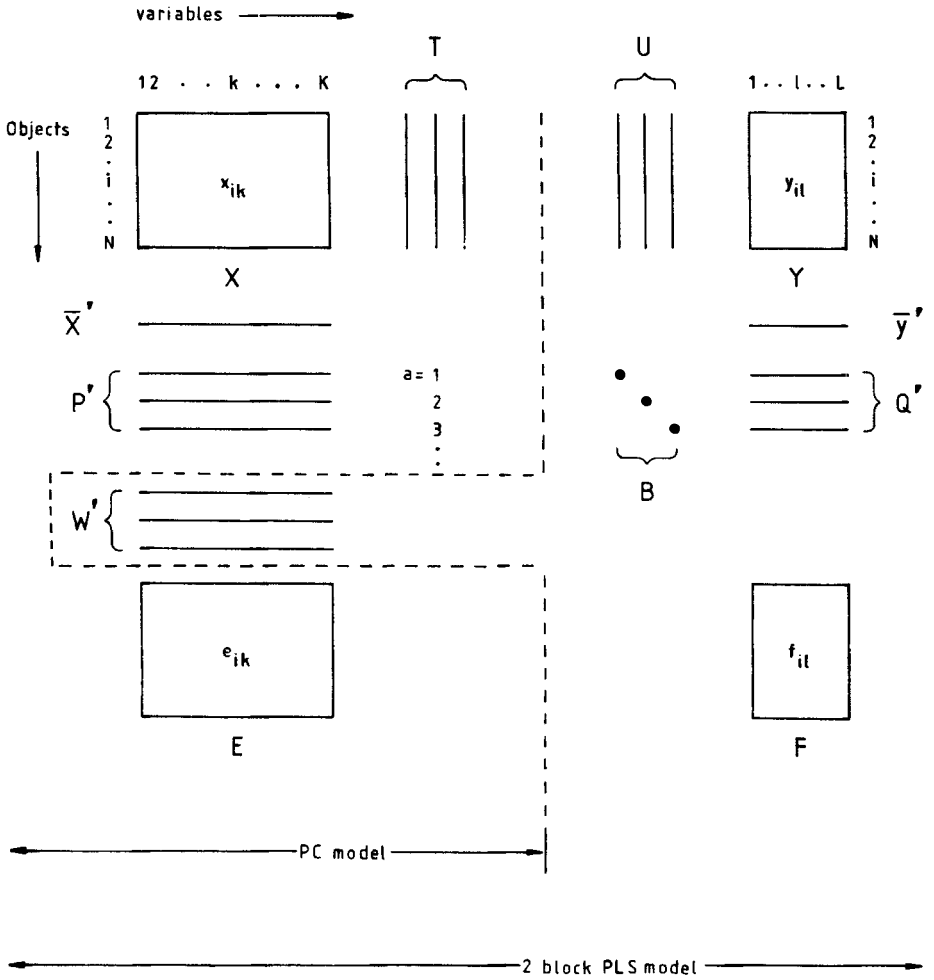

Figure 1. PCA of a matrix **X** and PLS analysis of the relation between two matrices **X** and **Y**

Computationally, PCA is usually handled by computing eigenvectors of $\mathbf{X}'\mathbf{X}$ (the covariance matrix) or $\mathbf{XX}'$ (the association matrix) or by the SVD algorithms originated by Golub.[6] If just the first few principal components are wanted, the NIPALS method of H. Wold[7] is advantageous because of its speed and simplicity.

Recently PCA has been generalized to cope with the analysis of a data matrix divided into blocks, the so-called PLS analysis of H. Wold.[8] PLS is an abbreviation for partial least squares modelling in latent variables *or* projection to latent structures. In chemistry, two-block predictive PLS (Figure 1) has been found useful for the multivariate calibration problem.[9-12] Thus, PLS gives an efficient way of predicting the y-values of new objects from their x-values, a generalization of multiple regression.

PLS analysis is based on extended NIPALS algorithms which decompose the block matrices into score matrices times loading matrices with the constraint that the score vectors of the same component are connected. For two-block PLS, this connection is by the inner relation (2c). For

the predictive two-block case:

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E} \tag{2a}$$

$$\mathbf{Y} = \mathbf{UQ}' + \mathbf{F} \tag{2b}$$

$$\mathbf{U} = \mathbf{TB} + \mathbf{H} \qquad (\mathbf{B} \text{ is diagonal}) \tag{2c}$$

The inner relation (2c) gives the predictive formulation for **Y**:

$$\mathbf{Y} = \mathbf{TBQ}' + \mathbf{F}^* \qquad (\mathbf{F}^* \text{ is not the same as } \mathbf{F} \text{ in (2b)}) \tag{2d}$$

Increasingly often, complex investigations give arrays of order three and higher (for chemical examples, see Table 1). The multivariate instrumentation developed in analytical chemistry is a strong driving force in this development, as discussed by Hirschfeld *et al.*[14] Satellite images, microscopic images, tomographic images and other images with several channels per pixel can also be seen as three-way data arrays. Multivariate time series analysis is a further problem area where the data may be structured as three- or four-way arrays. Hence, the need for methods able to cope with data arrays with more than two orders (ways) analogously to the multivariate analysis of ordinary two-way matrices has become urgent.

We use the terms 'ways', 'orders' and 'directions' for discussing the shape of the data arrays, not to confuse the readers and ourselves with several different meanings of 'dimension'. The latter word we shall use exclusively for the 'components' or 'factors', into which the arrays are decomposed. This is further discussed in Reference 15.

The directions of the data array represent sets of variables of *different types*. We note that most higher-order arrays are designed with one fundamental direction, in chemistry usually the common set of objects (e.g. analytical samples), characterized by the various types of variables (see Table 1).

Carroll *et al.*[16] have developed a canonical decomposition of a three-way array into a sum of vector products and more recently Young *et al.*[17] have developed a closely related non-metric variant. The notation is described later:

$$X = \sum_a \mathbf{t}_a \otimes \mathbf{u}_a \otimes \mathbf{v}_a \tag{3}$$

However, this decomposition lacks optimality properties; the generalization of PCA to three-way vector products is severely constrained and therefore mathematically difficult, as discussed by Lickteig.[18] Moreover, the extension of this approach to the PLS situation with X divided into blocks has not been found.

Table 1. Examples of data arrays of order three and higher, mainly from analytical chemistry

Order	Method(s)
3	Samples × liquid chromatography (LC) × UV spectrometry
3	Samples × excitation frequency × emission spectrometry (fluorescence)
3	Samples × gas chromatography (GC) × mass spectrometry (MS)
3	Two-dimensional NMR × third modification (e.g. pH, temperature)
4	Samples × chromatogram × column × spectrum, e.g. Reference 13
4	Samples × two-dimensional chromatography × UV spectrometry
6	[3]Image (three-dimensional grid samples) × time × LC × spectrometry

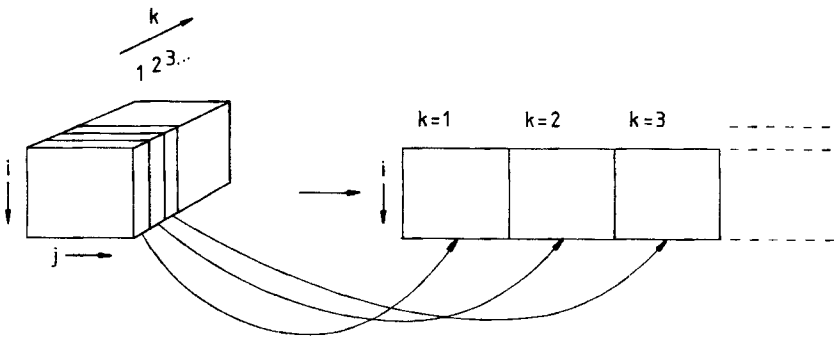

Figure 2. The unfolding of a three-dimensional data array X

Ho and co-workers,[19–21] Sanchez and Kowalski,[22] Lorber[23] and others have used so-called *rank annihilation* (RA) for the special case where the third direction in the array — corresponding to objects or chemical samples — has only two 'sheets'. RA has so far not been extended to the general case with several object 'sheets'.

In a little known paper, Lohmöller and H. Wold[24] show that a three-way array can be decomposed as a Kronecker product of a vector and a matrix, and a four-way array as a Kronecker product of a matrix and a matrix. They develop algorithms for principal components analysis (PCA) and some PLS models and show that the decomposition is equivalent to unfolding the arrays (Figure 2) to two-way matrices followed by ordinary PCA and PLS.

The same approach was used by Esbensen and Wold[25] who solved PCA and PLS of multidimensional arrays in a 'quick and dirty way' by unfolding the arrays to two-way matrices (Figure 2) which thereafter were analysed with standard NIPALS algorithms.

Sanchez and Kowalski[26] recently reported the decomposition of multiorder arrays by tensor calculus, which is closely related to the present treatment.

In the present article we use the optimality proof by Lohmöller and H. Wold[24] to develop NIPALS algorithms for decomposing a multiorder array of order R in terms of the product of a vector and an array of order $(R - 1)$. This gives a general approach to the analysis of such matrices which has some desirable statistical and numerical properties. This solution also has the appropriate form for calibration problems in chemistry and other branches of science, where one direction in the data arrays corresponds to objects, individuals, samples or process time points.

NOTATION AND NOMENCLATURE

We shall use lower case characters — $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\mathbf{t}$, $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, $\mathbf{p}$ and $\mathbf{q}$ — for column vectors, capitals such as $\mathbf{X}$ and $\mathbf{Y}$ for two-way matrices and italic capitals such as X and Y for arrays with three and more orders. We shall henceforth briefly use the terms 3-arrays, etc.; in general we refer to R-way arrays with $R > 2$ as R-arrays. A prime denotes a transpose of a vector or a 2-way matrix, i.e. $\mathbf{t}'$ is a row vector.

We distinguish between one direction in the arrays which relates to *objects* (chemical samples, cases, multivariate observations) and the other directions which all relate to *variables*. We use the index i for objects ($i = 1, 2, \ldots, N$) and the indices j, k, l, and m for variables, with the limits 1 and J, K, L or M, respectively. The object direction in the R-array is often referred to as columns in the array.

In image analysis the situation is reversed; one direction is variables (i) and the others (j, k, l and m) relate to objects (pixels).

SCOPE

In multivariate analysis we wish to separate the data into two parts — the systematic part and noise (residuals) — and moreover express the systematic part as one fraction relating only to objects times a second fraction relating only to variables, e.g. PCA and equation (1) above.

Usually one also wants to achieve further objectives, such as the noise part being as small as possible (least squares) and that the systematic part have certain properties, such as a high correlation between blocks (PLS and canonical correlation) and high discrimination between objects from different classes (canonical variates and discriminant analysis).

The simplest way to achieve this decomposition for an R-array is to let one direction — usually the one corresponding to 'objects' or cases — be expressed as vectors and the other directions together be expressed as an array of order $R-1$. For the case of a 3-array X:

$$X = \mathbf{t}_1 \otimes \mathbf{P}_1 + \mathbf{t}_2 \otimes \mathbf{P}_2 + \ldots + E = \mathbf{T}. \otimes P + E \tag{4}$$

We note that if $R = 2$, $\mathbf{X}$ is a usual 2-way matrix and $\mathbf{P}$ is a one-way matrix, i.e. a vector. This then gives the ordinary decomposition into products of two vectors.

If now the vectors $\mathbf{t}_i$ are orthogonal to each other as well as the matrices $\mathbf{P}_i$ being orthogonal to each other (see below), and $\mathbf{T}. \otimes P$ is a least squares model of X for any number of 'components', a, this decomposition is a direct generalization of the PC decomposition. If we analogously decompose the R-array Y with the same 'score matrix' $\mathbf{T}$, we have the R-way analogue to equation (2), i.e. the PLS analysis of a two-block R-array.

As we shall see below, the order, R, need not be the same in the $\mathbf{X}$-part and the $\mathbf{Y}$-part of the PLS model generalization. In fact, this follows already from the PLS model with a 2-array $\mathbf{X}$ and a 1-array $\mathbf{Y}$, which is a special case of two-block 2-array PLS.[27]

PLS and PCA in combination incorporate the regression problem, discriminant analysis, canonical variates, canonical correlation, procrustean rotation and other seemingly disparate problems. This has been shown by H. Wold *et al.*,[28] Lohmöller and Wold,[24] S. Wold *et al.*[29] and Martens.[12] Hence, when PCA and PLS are generalized to 3-arrays and higher, this directly gives generalizations of these other analyses also.

EXAMPLES

We use two examples to illustrate the methods. The first is a small numerical example with a three-way array X with $N = 3$, $J = K = 2$ and a two-way matrix $\mathbf{Y}$ ($N = 3$, $L = 2$) (see Table 2). This example is used to show the results of the algorithms.

The second example has real data from an experiment of multivariate calibration with liquid chromatography and a UV array detector. Each analysed sample (object) is either a mixture of the chemical constituents anthracene and phenanthrene or a pure sample of one of these constituents. The sample was dissolved in methanol. The conditions are the same as used previously[30] with the eluent composition being 95% methanol and 5% water. The known concentrations of the two constituents in the $N = 6$ samples are contained in a 6×2 matrix $\mathbf{Y}$ for the calibration. A test set of four samples with known amounts of the constituents was kept aside to check the prediction power of the calibrated model.

Table 2. Data set for the small numerical example. The two layers of $X(k = 1$ and $k = 2)$ are written beside each other, which makes the data and results appear in unfolded form. the x-values for the three training set objects are all 3, 4 or 5 divided by $\sqrt{50}$. The data are used in *unscaled and uncentred form*

	$k = 1$		$k = 2$			
i	x_{i11}	x_{i21}	x_{i12}	x_{i22}	y_{i1}	y_{i2}
1	0·424264	0·565685	0·565685	0·424264	1·0	1·0
2	0·565685	0·424264	0·424264	0·565685	2·0	1·5
3	0·707101	0·707101	0·707101	0·707101	3·0	2·0
test	0·5	0·6	0·6	0·4	?	?

MATRIX NOTATION FOR R-ARRAYS

We need some new notation and conventions for R-way matrix algebra. First some preliminaries with ordinary two-way matrices and one-way vectors.

Product of a row vector and a matrix, typically $\mathbf{p}' = \mathbf{t}'\mathbf{X}$:

$$p_j = \sum_i t_i x_{ij} \tag{5}$$

Product of a matrix and a column vector, tyically $\mathbf{t} = \mathbf{X}\mathbf{p}$:

$$t_i = \sum_j x_{ij} p_j \tag{6}$$

Outer product between a column and a row vector, typically $\mathbf{X} = \mathbf{t}\mathbf{p}'$:

$$x_{ij} = t_i p_j \tag{7}$$

Hadamard product between two matrices of same size, $\mathbf{C} = \mathbf{A} * \mathbf{B}$:

$$c_{ij} = a_{ij} b_{ij} \tag{8}$$

Below follow some definitions, which refer to 3-arrays. The extension to 4-arrays, 5-arrays, etc. is evident.

Generalized transpose of an R-array:

$$X'' \text{ (G-transpose)} \tag{9}$$

This is needed for various generalized matrix products. This G-transpose means that one, usually the first, direction in the array is transposed, but the others are left intact. This transpose has the same properties as the ordinary transpose with respect to a product, i.e. if $\mathbf{t} = X \,..\, \mathbf{W}''$, then $\mathbf{t}' = \mathbf{W} \,..\, X''$.

In the generalized transpose of an $(R - 1)$-array (e.g. 2-array as $\mathbf{W}''$ above) *none* of the directions is transposed. This is needed to obtain a consistent product of an R-array with an $(R - 1)$-array.

The *norm* of an R-array $\| X \|$ is defined as the square root of the sum of the squares of all the elements in the array.

Generalized Hadamard product, typically $Z = X * Y$:

$$z_{ijk\ldots} = x_{ijk\ldots} y_{ijk\ldots} \tag{10}$$

Kronecker product between a vector and an array, $X = \mathbf{t} \otimes \mathbf{P}$ (or P):

$$x_{ijk} = t_i p_{jk} \tag{11}$$
$$x_{ijkm} = t_i p_{jkm}$$

and so on.

$(R-1)$-way product of two R-way matrices, typically $\mathbf{A} = X \,..\, X''$.

$$a_{ij} = \mathbf{X}_i * \mathbf{X}_j \tag{12}$$

i.e.

$$a_{ij} = \sum_{klm\ldots} x_{iklm\ldots} x_{jklm\ldots}$$

In-out-product of a 2-matrix and an R-array, typically $X = \mathbf{T}. \otimes P$:

$$x_{ijk\ldots} = \sum_a t_{ia} p_{ajk\ldots} \tag{13}$$

Row vector times an R-array, typically $\mathbf{P} = \mathbf{t}' X$:

$$p_{jk\ldots} = \sum_i t_i x_{ijk\ldots} \tag{14}$$

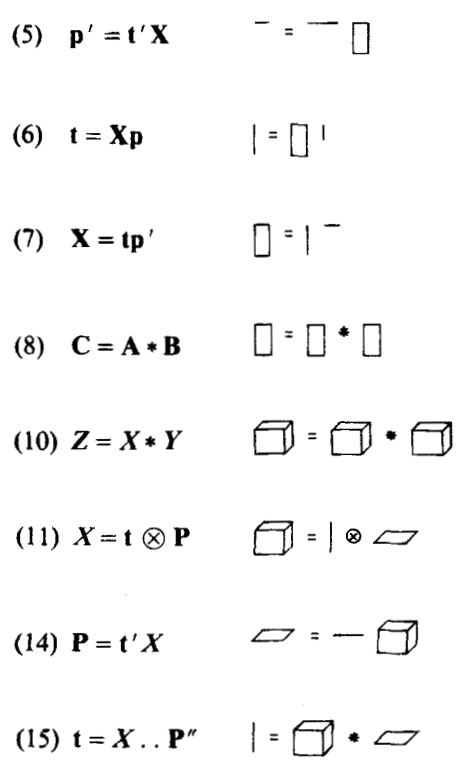

Figure 3. An effort to clarify the R-way matrix notation

R-array times $(R-1)$-array, typically $\mathbf{t} = X \,..\, \mathbf{P}''$:

$$t_i = \sum_{jk\ldots} x_{ijk\ldots} p_{jk\ldots} \tag{15}$$

i.e.

$$t_i = \mathbf{X}_i * \mathbf{P}$$

With this notation summarized in Figure 3 we shall now develop the generalized PC and PLS models and algorithms for their estimation.

GENERALIZED PC AND PLS MODELS

The PC model becomes

$$X = \mathbf{T}. \otimes P + \mathbf{E} \tag{16}$$

For a single component, this becomes (assuming that X is three-way):

$$X = \mathbf{t} \otimes \mathbf{P} = E \tag{17}$$

The two-block predictive PLS model is

$$X = \mathbf{T}. \otimes P + E \tag{18a}$$

$$Y = \mathbf{U}. \otimes Q + F \tag{18b}$$

$$\mathbf{U} = \mathbf{TB} + \mathbf{H} \qquad (\mathbf{B} \text{ is diagonal}) \tag{18c}$$

This inner relation inserted into (18b) gives the predictive relation for Y:

$$Y = \mathbf{T}. \otimes Q\mathbf{B} + \mathbf{F}^* \tag{18d}$$

Note that the orders of X and Y need not be the same; Y may have a lower order than X. This difference is then reflected in the different orders of on the one hand P and W and on the other hand Q, respectively. W is explained below in the PLS algorithm.

If it is so desired in a particular application, the 'loading arrays' P and Q can, sheet-wise (for each component), be further decomposed in terms of principal components, say $\mathbf{P}_a = \mathbf{U}_a\mathbf{R}_a$. For a 3-array, two-component PC model, for example

$$X = \mathbf{t}_1\mathbf{U}_1\mathbf{R}_1 + \mathbf{t}_2\mathbf{U}_2\mathbf{R}_2 + E \tag{19}$$

In a simple case, $\mathbf{P}_1$ and $\mathbf{P}_2$ may have rank one, which then makes the model reduce to the Carroll CANDECOMP[16] and the Ho *et al.* rank annihilation[19] models.

ALGORITHMS

We present below the NIPALS algorithms for R-arrays. They usually converge in about 10 iterations, which make them fast, in particular since they work directly on the raw data matrices. Their equivalence to the ordinary two-way algorithms makes it possible to use standard two-way software.

The three-way PCA algorithm was first developed by Lohmöller and H. Wold.[24] We present it here to point out the obvious generalization to four and more directions in the multi-way arrays. The PLS algorithm is the result of a combination of the Lohmöller–Wold decomposition and the multidimensional PLS–NIPALS algorithm of Wold *et al.*[5,10]

The solution may be constrained by forcing the PC loading arrays $\mathbf{P}_a$ and PLS weight arrays $\mathbf{W}_a$ to each take the form of the outer product of two vectors. Alternatively, geographical or other information may be used to smooth the solution, as outlined by Esbensen and Wold[25] in their SPACE method.

The PCA NIPALS algorithm

The data set-up and the resulting decomposition in one component is shown in Figure 4. The algorithm goes as follows:

(A) Preprocessing, Usually, the array **X** is scaled to unit variance by dividing each element by the column standard deviation $s_{jk\ldots}$ (calculated over objects). Then the array may be centered by subtracting the column mean from each element. This is analogous to the two-way case. Many prefer that the centering precedes the scaling, but this is arbitrary.

(a 1) $x_{ijk\ldots} = (x_{ijk\ldots}/s_{jk\ldots}) - \bar{x}_{jk\ldots}$

(B) Initialize the component index, a:

(a 2) $a = 0$

(C) Then, for each component:

(a 3) $a = a + 1$

(a 4) **t**-start = column in X with largest variance.

(a 4.1) **t** is normalized to $\| \mathbf{t} \| = 1$.

(a 5) $\mathbf{P} = \mathbf{t}'X$ (see equation (14)).

(a 5.1) Alternatively to (a 4.1), **P** can be normalized; say normalized so that $\| \mathbf{P} \| = 1$.

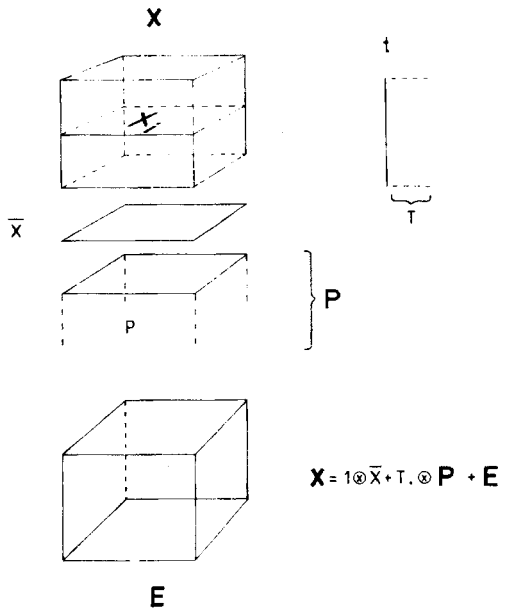

Figure 4. Three-dimensional data array and its generalized principal components decomposition

(a 5.2) Here an additional NIPALS loop may be inserted that decomposes **P** into, say, $\mathbf{ur}'$, or $\mathbf{u}_1\mathbf{r}_1' + \mathbf{u}_2\mathbf{r}_2'$. This loop should then be run through twice, i.e. for the simplest case $\mathbf{P} = \mathbf{ur}'$ (**u**-start is 'largest' column in **P**):

(0) $h = 0$
(i) $\mathbf{r}' = \mathbf{u}'\mathbf{P}/\mathbf{u}'\mathbf{u}$
(ii) norm to $\| \mathbf{r} \| = 1$
(iii) $\mathbf{u} = \mathbf{Pr}$
(iv) $h = h + 1$; if $h < 2$ then back to (i)
(v) set $\mathbf{P} = \mathbf{ur}'$ and proceed with (a 7).

(a 7) $\mathbf{t} = X \,..\, \mathbf{P}''/\| \mathbf{P} \|^2$ (see equation (15)).

(a 8) Check convergence: back to (a 5) if

$$d = (\mathbf{t}_{new} - \mathbf{t}_{old})'(\mathbf{t}_{new} - \mathbf{t}_{old})/(N\mathbf{t}_{new}'\mathbf{t}_{new}) > 10^{-10}$$

(a 9) After convergence, residuals: $E = X - \mathbf{t} \otimes \mathbf{P}$.

(D) For the next component, set $X = E$ and return to a 3.

(E) To estimate the adequate number of components, use cross-validation in analogy with the 2-array case according to Wold,[31] or standard criteria such as stopping when sufficiently much of the variance in X has been described or when the variance of **P** goes below 1·0 (if **t** is normalized to length one; when **P** is normalized, one looks instead at the variance of **t**).

(F) The **t**-values for new objects (test objects) are calculated as follows:

(i) Scale and center the object array X using (a 1) above.
(ii) Calculate $\mathbf{t}_a$ for all pertinent model dimensions ($a = 1, \ldots, A$) as:

(a 10) $\mathbf{t}_a = X * \mathbf{P}_a$

The size of the residuals E after the last **t**-value can be used in the ordinary SIMCA-MACUP way to check if the new object is similar to the class training set or not.

The predictive PLS two-block, mode A NIPALS algorithm

The data set-up and the resulting decomposition in one component are shown in Figure 5. The algorithm goes as follows.

(A) Preprocessing. Usually, the arrays X and Y are scaled to unit variance by dividing each element by the column standard deviation $s_{jk\ldots}$. Then, the array is often centered by subtracting the column mean from each element. This is analogous to the two-way case. Below in (b 1), z denotes x or y. Centering often precedes scaling, but the order is arbitrary:

(b 1) $z_{ijk\ldots} = (z_{ijk\ldots}/s_{jk\ldots}) - \bar{z}_{jk\ldots}$

(B) Initialize the component index, a:

(b 2) $a = 0$

(C) Then, for each component:

(b 3) $a = a + 1$
(b 4) **u**-start = column in Y with largest variance.
(b 5) $\mathbf{W} = \mathbf{u}'X/\mathbf{u}'\mathbf{u}$ (see equation (14)).
(b 6) Normalize so that $\| \mathbf{W} \| = 1$, or alternatively in step (b 7), normalize to $\| \mathbf{t} \| = 1$.
(b 6.1) Here we may, in analogy with steps (a 5.2) above, decompose **W** into, say, $\mathbf{sr}'$ with an extra NIPALS shunt.

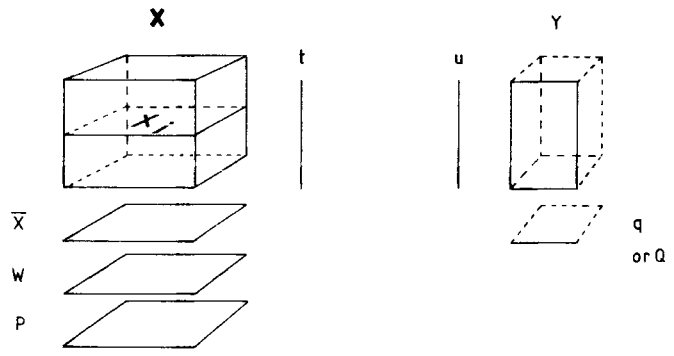

Figure 5. Three-dimensional data array X and a data array Y connected by a two-block PLS-model with one component

(b 7) $\mathbf{t} = X \,..\, \mathbf{W}''/\| \mathbf{W} \|^2$ (see equation (15)).

(b 8) Compute the convergence criterion:

$$d = (\mathbf{t}_{new} - \mathbf{t}_{old})'(\mathbf{t}_{new} - \mathbf{t}_{old})/(N\mathbf{t}_{new}'\mathbf{t}_{new})$$

(b 9) $Q = \mathbf{t}'Y/\mathbf{t}'\mathbf{t}$ (see equation (14)).

(b 10) Convergence? Go to (b 12) if d (from b 8) $< 10^{-10}$.

(b 11) $\mathbf{u} = Y \,..\, \mathbf{Q}''/\| \mathbf{Q} \|^2$ (see equation (15)).
Back to (b 5).

(b 12) After convergence:

(i) X-loadings:

$\mathbf{P} = \mathbf{t}'X/\mathbf{t}'\mathbf{t}$ (see analogy in step b 9)

We note that we *cannot* decompose $\mathbf{P}$ into a few product terms $\mathbf{u}_b\mathbf{v}_b'$ because then the score vectors $\mathbf{t}_a$ no longer become orthogonal.

If we wish to have $\mathbf{P}$ normalized to $\| \mathbf{P} \| = 1$, then that is done here by dividing $\mathbf{P}$ by, say, $\mathbf{c} = \| \mathbf{P}_{old} \|$. Then $\mathbf{t}$ and $\mathbf{W}$ must be multiplied by the same scalar, $\mathbf{c}$, to retain the numerical equivalence of the solution.

(ii) (Normalize $\mathbf{Q}$ to one). Not necessary. See iii.

(iii) Compute b_a in the inner relation ($\mathbf{u}_a = b_a\mathbf{t}_a + \mathbf{h}$):

$$b_a = \mathbf{t}_a'\mathbf{u}_a/\mathbf{t}_a'\mathbf{t}_a$$

Note that if $\mathbf{Q}_a$ is not normed b_a will be 1·0.

(b 13) Residuals:

(X) $E = X - \mathbf{t} \otimes \mathbf{P}$

(Y) $F = Y - \mathbf{t} \otimes \mathbf{Q}b_a$

(D) For the next component, set $X = E$ and $Y = F$ and return to b 3.

(E) To estimate the adequate number of components, use cross-validation in analogy with the two-way case[31] or an analysis of variance on the inner relation ($\mathbf{u} = \mathrm{b} \,.\, \mathbf{t} + \mathbf{h}$),[32]

(F) Predictions of Y for new objects (test objects) are made as follows:

(i) Scale and center the object array X using (b 1) above.

(ii) Calculate the first t-value by (b 7).

(iii) Calculate X-residuals from (b 13X).

(iv) Calculate the second t-value by (b 7).
(v) X-residuals, etc. Go on until all pertinent t-values have been calculated (for dimension $1, 2, \ldots, A$).
(vi) Calculate predicted Y-values as

(b 14) $Y = \mathbf{t}_1\mathbf{b}_1\mathbf{Q}_1 + \mathbf{t}_2\mathbf{b}_2\mathbf{Q}_2 + \ldots$

The Y-values are then obtained in scaled and centred form, which can be converted to the original co-ordinates by using the transformation (b 1) backwards.

The size of the X-residuals E after the last t-value can be used in the ordinary SIMCA-MACUP way[4,5,29] to check if the new object is similar to the calibration set or not.

SOME PROPERTIES OF THE ALGORITHMS

If (a 5) is substituted into (a 7) we obtain (c is a scalar):

$$\mathbf{t} = cX \ldots X''\mathbf{t} \tag{20}$$

Hence $\mathbf{t}$ is an eigenvector of the generalized association matrix, and the NIPALS algorithm a variant of the power method. This proves the convergence of the algorithm unless it happens to start exactly on another eigenvector, which is extremely unlikely if t-start is chosen as the X-column with the largest variance.

If we merge the indices $jk \ldots$ into a single index, say r, we see that the algorithm is equivalent to the ordinary two-way algorithms with X unfolded to a 2-array.[24,25] This is another way to prove the eigenvector — least squares — properties of the solution. For the generalized PC solution, we may call $\mathbf{t}$ and $\mathbf{P}$ singular vectors and matrices of X.

By inserting (b 5), (b 9) and (b 11) into (b 7) it is easy to prove that for the first dimension the PLS vector $\mathbf{t}$ is an eigenvector of $X \ldots X'' \; Y \ldots Y''$ and $\mathbf{u}$ an eigenvector of the generalization of $\mathbf{YY'XX'}$. For later dimensions, $\mathbf{t}$ and $\mathbf{u}$ are instead eigenvectors of $E \ldots E'' \; F \ldots F''$, etc.

Also here we can merge indices and obtain the equivalence to the solution obtained by unfolding.

From this it follows that the PC loading array P has layers which are orthogonal to each other ($\| \mathbf{P}_a * \mathbf{P}_b \| = 0$ if $a \neq b$) and so has the PLS weight array W. The t-vectors of both the PC and the PLS models are orthogonal to each other.

Finally, we realize that the above decompositions and algorithms are valid for arrays with orders $3, 4, 5, \ldots$, i.e. for any finite order.

THE RANK OF AN R-ARRAY

We can now define the effective rank of the R-array X as the number of 'components — generalized PC model dimensions $\mathbf{t}_a \otimes \mathbf{P}$ — needed to make the norm of the residuals, $\| E \|$, sufficiently small for a given purpose. We note that this rank is different from that discussed by Lickteig,[18] who is concerned only with (vector × vector × vector) decompositions.

The number of components determined by cross-validation relates to the predictive ability of the model with a certain fraction of the data temporarily deleted. This is a lower bound to the 'effective' rank, which may be too low in some instances. In multivariate calibration, experience shows[12] that, if the numbers of variables and calibration samples so permit, one should add a few PLS components beyond the cross-validation limit.

SMALL NUMERICAL EXAMPLE

The PCA of the X-data in Table 2 shows the first component to be dominating and the third to be very small (see Table 3). Consequently, the PLS analysis gives almost the same decomposition of the X-part as the PCA (Table 3). In Table 4 the score values $\mathbf{t}_a$ for the PC and PLS analyses are shown together with the predicted y-values for the objects 1–4 by the PLS model with two components.

Table 3. Resulting loadings (PC and PLS) and PLS weights for the data in Table 2. SS denotes remaining sum of squares. **Q** has not been normalized; hence $b_a = 1{\cdot}0$ in (b 12, iii)

		SS-**X**	SS-**Y**	11	21	12	22	*y*1	*y*2
	$a=0$	4·0	21·25						
PCA,	$a=1$	0·040		0·5	0·5	0·5	0·5		
PCA,	$a=2$	4×10^{-14}		0·5	−0·5	−0·5	0·5		
PLS,	$\mathbf{P(Q)}_1$	0·040	1·0	0·5	0·5	0·5	0·5	1·82	1·34
	$\mathbf{P(Q)}_2$	4×10^{-14}	0·386	0·49	−0·51	−0·51	0·49	3·50	1·74
PLS,	$\mathbf{W}_1$			0·51	0·49	0·49	0·51		
	$\mathbf{W}_2$			0·49	−0·51	−0·51	0·49		

Table 4. Resulting scores $\mathbf{t}_a$ (and $\mathbf{u}_a$ for PLS) for the PC and PLS analyses of the data in Table 2, together with predicted y-values from the two-component PLS model (see Table 3)

	PC		PLS			PLS predictions		
i	*t*1	*t*2	*t*1	*t*2	*u*1	*u*2	*y*1	*y*2
1	0·9899	−0·1414	0·9873	−0·1416	0·6181	−0·2196	1·303	1·076
2	0·9899	0·1414	0·9823	0·1413	1·1053	0·0633	2·303	1·576
3	1·4142	0·0000	1·4140	−0·0002	1·5925	0·1089	2·576	1·894
4	1·0000	−0·0100	0·9983	−0·1002			1·468	1·162

EXAMPLE 2: MULTIVARIATE CALIBRATION WITH HPLC-UV DATA

As an experimental data set, data from liquid chromatography with UV detection was chosen. There were six calibration samples and four test samples. They were mixtures of anthracene and phenanthrene with overlap in both the spectral and the chromatographic parts.[30] For the illustration, ten wavelengths were selected from the 180 available and only ten points were chosen from the time axis. This selection was made to fit the data set into available programs. In real applications, of course, larger data sets would be used.

The resulting X-block with calibration data is a $6\times10\times10$ three-way matrix. The Y-block for PLS is a 6×2 two-way matrix. The results for the complete data set will all wavelengths and chromatographic data will be published separately.

First a principal component analysis was carried out on only the X (three-way) matrix. This showed that three components were able to describe 99·6% of the remaining sum of squares after variance scaling and subtraction of the mean.

The PLS modelling was carried out without and with constraints on the weight matrices $\mathbf{W}_a$. In the latter case, $\mathbf{W}_a$ was forced to rank 1 with the decomposition $\mathbf{W} = \mathbf{cs}'$. The results of the PLS calibration for these data are given in Table 5.

Table 5. The PLS model of properties for the experimental data. SS_Y is the sum of squares of the *y*-variables. 'Normal' refers to non-constrained **W** and 'constrained' — to constrained **W**, respectively. *F* inner relation refers to the *F*-statistics of the inner PLS relation. The last column refers to the PCA of the **X**-part of the data

	SS_Y		F inner relation		PCA
Comp. no	normal	constrained	normal	constrained	**X**-variance left
1	38%	38%	148	149	16%
2	1%	1%	143	145	7%
3	0·02%	0·01%	616	988	0·36%
4	0·002%	0·006%	64	8	0·001%

The table shows that after variance-scaling and centring, three components describe 99·983% of the sum of squares of the *Y*-block and that a further modelling is not necessary. The drop in the *F* on the inner relation confirms this fact. The number of objects is too small to allow cross-validation.

The constrained PLS analysis gives a slightly better model with the information a little better concentrated in the first three components.

Predictions were carried out on the four test samples with the three-component model. The results are shown in Table 6.

These results are good, considering the amount of overlap of chromatograms and spectra and the fact that only ten wavelengths and ten time samples in the chromatograms were used. The worst predictions are found for the low anthracene concentrations. It should be noted here that these results can be obtained with unfolding and the existing SIMCA programs.

The differences between the constrained model predictions and those of the unconstrained model are small.

Table 6. True and predicted results (ppm) for the test samples

	Anthracene			Phenanthrene		
	Predicted			Predicted		
Sample	Normal	Constrained	True	Normal	Constrained	True
1	9·11	8·55	8·04	14·8	15·1	14·4
2	11·0	10·8	8·16	29·0	29·1	28·9
3	17·7	17·8	16·0	39·3	39·5	40·9
4	23·6	23·1	23·7	29·5	29·8	28·6
Predicted SS	12·2	10·8		3·6	3·9	

CONCLUSIONS AND DISCUSSION

The decomposition of a multiorder (*R*-way) array into the product of a vector and an $(R-1)$-way array is a straightforward, almost trivial, extension of the ordinary PC and PLS decompositions of two-way arrays.

The formalism makes it easy to understand why a three-way array in the general case *cannot*

Table 7. Decompositions of a four-way array based on the properties of topological and measure spaces and their mapping operators

$4 = 1 \otimes 1 \otimes 1 \otimes 1$	extremely difficult
$4 = 1 \otimes 1 \otimes 2$ (or $1 \otimes 2 \otimes 1 \ldots$)	difficult
$4 = 1 \otimes 3$ (or $3 \otimes 1$)	OK
$4 = 2 \otimes 2$	OK

be decomposed into products of three vectors. This can be made only if the loading matrices of the generalized PC decomposition have exactly the rank one. This may happen in a chemical application of, say, spectroscopy times chromatography times samples — like the one used as example 2 — when there are no interactions between the constituents and the detector is exactly linear, i.e. Beer's model is exactly valid.[32,33] Though Beer's model is often a rather good approximation, it is rarely 'exact'. With the present formalism it is possible to calculate an optimal decomposition and then, thereafter, further decompose the loading matrices and investigate their approximate rank, how much that is explained by its first component, etc.

Lohmöller and H. Wold[24] decompose four-way matrices into Kronecker products of two-way matrices; $X = \Sigma_a \mathbf{P}_a \otimes \mathbf{Q}_a$. This may be warranted in some cases, but we find it more practical to keep one direction in the 4-array or ($R > 4$)-array distinct and express that as a vector. This makes it possible to identify this direction with an ordinary latent variable which can carry information to other blocks in path models, as exemplified in the PLS multivariate calibration example above. This also makes it possible to use ordinary score plots for visualizing relations between objects when the data for each object have order two or higher. The possible decompositions of a four-way table are shown in Table 7.

Multiple regression, multivariate analysis of variance, canonical correlation, discriminant analysis, canonical variates, procrustean rotations, pattern recognition, time series analysis and other multivariate data analytical methods can all be approached by PLS modelling. This makes the present methodology have even further generality, since all these methods are now easily generalized to R-way arrays via PLS and PC modelling.

The NIPALS algorithms are fast and efficient as long as we are interested only in the few first dimensions in the expansions of X (and Y in the PLS case). This makes them useful even on personal computers, in particular since they are simple to program and available in commercial packages for those less interested in writing their own software.

Since three-way and four-way data arrays are becoming common in science and technology (see introduction), the generalized PC and PLS expansions may have many applications. Many of these comply with the design with a common set of objects characterized by ($R - 1$) directions of variables. We can also envisage similar generalizations of the 'transposed PLS design', where a common set of variables is measured on different sets of objects related in a causal way. One simple illustration of this concept is that of toxic substances measured in a set of ambient air samples and in a set of blood samples in a study of pollution methabolic pathways.

Table 1 lists a series of potential chemical applications of R-way decomposition with R ranging from 3 to 6. The conceptual problems of interpretation and visualization of the higher order applications may turn out to be much more difficult than the mere implementation of PCA or PLS algorithms.

Finally, we note that the number of data elements in a multiorder array is often very large. Hence, in order to analyse such arrays in practice, efficient methods are needed for reducing the data volumes to manageable size. Such methods are not yet developed and should therefore be given high priority to make the information in multiorder arrays easier available.

ACKNOWLEDGEMENTS

We are grateful for support from the Swedish Natural Science Research Council (NFR), the Swedish Board for Technical Development (STU), the Swedish Council for Planning and Coordination of Research (FRN), the Peat Research Project in Umeå and the Norwegian Computing Center. Herman Wold's kind advice has been invaluable as always.

REFERENCES

1. R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations*, Wiley, New York (1977).
2. A. Höskuldsson, in *Proc. Symp. Appl. Statist.*, NEUCC, Copenhagen January 1984, ed. by L. S. Mortensen and A. Höskuldsen, ISSN 0105 9653.
3. K. Mardia, J. Kent and J. Bibby, *Multivariate Analysis*, Academic Press, London (1979).
4. C. Albano, W. Dunn, U. Edlund, E. Johansson, B. Norden, M. Sjöström and S. Wold, *Anal. Chim. Acta*, **103**, 429 (1978).
5. S. Wold, C. Albano, W. J. Dunn, K. Esbensen, S. Hellberg, E. Johansson and M. Sjöström, in *Food Research and Data Analysis*, ed. by H. Martens and H. Russwurm, Applied Science Publishers, London (1983).
6. G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Oxford (1983).
7. H. Wold, in *Research Papers in Statistics*, ed. by F. David, pp. 411–444, Wiley, New York (1966).
8. H. Wold, in *Systems under Indirect Observation* ed. by H. Jöreskog and H. Wold, Vol 2, pp. 1–54, North-Holland, Amsterdam (1982).
9. W. Lindberg, J. Persson and S. Wold, *Anal. Chem.* **55**, 643 (1983).
10. S. Wold, H. Martens and H. Wold, in *Matrix Pencils*, ed. by A. Ruhe and B. Kågström, pp. 286–293, Lecture Notes in Mathematics, Springer, Heidelberg (1983).
11. H. Martens and T. Naes, *Trends in Analytical Chemistry* **3**, 204 and 266 (1984).
12. H. Martens, 'Multivariate calibration', *Ph.D. Thesis*, Technical University of Norway, Trondheim (1985).
13. J. Burger, S. Ramos and B. R. Kowalski, *Anal. Chem.* **58**, 57 (1986).
14. T. Hirschfeld, J. B. Callis and B. R. Kowalski, *Science* **226**, 312 (1984).
15. K. Esbensen, P. Geladi and S. Wold, 'Relationships between data array configurations, problem formulation and conceptual models in data analysis', in preparation, (1986).
16. J. Carroll, S. Pruzansky and J. Kruskal, *Psychometrika* **45**, 3 (1980).
17. F. Young, Y. Takane and J. de Leeuw, *Psychometrika* **43**, 279 (1978).
18. T. Lickteig, *Linear Algebra and its Applications* **69**, 95 (1985).
19. C. Ho, G. Christian and E. Davidson, *Anal. Chem.* **50**, 1108 (1978).
20. C. Appellof and E. Davidson, *Anal. Chim. Acta* **146**, 9 (1983).
21. C. Ho, G. Christian and E. Davidson, *Anal. Chem.* **53**, 92 (1981).
22. E. Sanchez and B. Kowalski, *Anal. Chem.* **58**, 499 (1986).
23. A. Lorber, *Anal. Chim. Acta.* **164**, 293 (1984).
24. J. Lohmöller and H. Wold, presented at the *European Meeting of the Psychometrics Society*, Groningen, Netherlands, June 1980.
25. K. Esbensen and S. Wold, in *Proc. Symp. Applied Statistics*, Stavanger, Norway, June 1983, pp. 11–36.
26. E. Sanchez and B. R. Kowalski, *Anal. Chim. Acta.* (1986).
27. S. Wold, A. Ruhe, H. Wold and W. J. Dunn, *SIAM J. Sci. Stat. Comput.* **5**, 735 (1984).
28. H. Wold *et al. Third Congr. Econometrics*, Toronto 1975; Uppsala University Statistics Dept. Research rep. 75 : 4.
29. S. Wold *et al.*, in *Proc. NATO Adv. Study Inst. on Chemometrics*, Cosenza, Italy, 1983, ed. by B. R. Kowalski, pp. 17–95, Reidel Publ. Co., Dordrecht, Holland (1984).
30. W. Lindberg, J. Ohman and S. Wold, *Anal. Chem.* **58**, 299 (1986).
31. S. Wold, *Technometrics* **20**, 397 (1978).
32. P. Geladi and B. R. Kowalski, *Anal. Chim. Acta* **185**, 1 (1986).
33. P. Geladi and B. R. Kowalski, *Anal. Chim. Acta* **185**, 19 (1986).