JOURNAL OF CHEMOMETRICS, VOL. 10, 47–61 (1996)

MULTIWAY CALIBRATION. MULTILINEAR PLS

RASMUS BRO

Food Technology, Royal Veterinary and Agricultural University, Thorvaldsensvej 40, 6, ii, DK-1871 Frederiksberg C, Denmark

SUMMARY

A new multiway regression method called *N*-way partial least squares (N-PLS) is presented. The emphasis is on the three-way PLS version (tri-PLS), but it is shown how to extend the algorithm to higher orders. The developed algorithm is superior to unfolding methods, primarily owing to a stabilization of the decomposition. This stabilization potentially gives increased interpretability and better predictions. The algorithm is fast compared with e.g. PARAFAC, because it consists of solving eigenvalue problems.

An example of the developed algorithm taken from the sugar industry is shown and compared with unfold-PLS. Fluorescence excitation–emission matrices (EEMs) are measured on white sugar solutions and used to predict the ash content of the sugar. The predictions are comparable by the two methods, but there is a clear difference in the interpretability of the two solutions. Also shown is a simulated example of EEMs with very noisy measurements and a low relative signal from the analyte of interest. The predictions from unfold-PLS are almost twice as bad as from tri-PLS despite the large number of samples (125) used in the calibration.

The algorithms are available from World Wide Web: hhtp:\\newton.foodsci.kvl.dk\foodtech.

KEY WORDS multilinear PLS; multiway calibration; unfold methods

INTRODUCTION

Partial least squares regression is a method for building regression models between independent (called x) and dependent variables (called y). The algorithm works on first-order data, i.e. for each sample/object several variables are determined. The set of calibration samples can subsequently be arranged in two matrices, one containing the independent and one the dependent variables. Each row contains the determined variables of one sample.

PLS is based on a bilinear decomposition of the calibration matrix. Many reviews explain the PLS algorithm from a mathematical, geometrical and statistical view respectively,[1–5] and the basic algorithm will not be explained in detail in this paper.

The most important feature of PLS is that the decomposition is accomplished such that the successively computed score vectors have the property of maximum covariance with the unexplained part of the dependent variable. This is mainly where PLS differs from principal component regression (PCR).

In this paper PLS regression is extended to multiway data, with the main emphasis on three-way data. Three-way data can occur when the variables are characterized by a matrix (second-order data) instead of a vector. Examples could be an excitation–emission matrix (EEM), the

CCC 0886–9383/96/010047–15

Received 13 March 1995
Accepted 3 July 1995

output from an HPLC in conjunction with a multiwavelength detector or a certain set of variables measured on several occasions or locations.

It is worthwhile noticing that there are many ways to understand the meaning of (multi)linearity in these types of calibration models. PLS and similar models actually consist of two steps: decomposition of the calibration array and establishment of a relation (regression) between the decomposed array of independent variables and the dependent variable(s). Feedforward neural networks[6] are often used as non-linear calibration models in many sciences. These methods, however, are most often still bilinear in their decomposition of the calibration matrix. It is in the relation between the dependent variable and the decomposed calibration matrix that the non-linearity is introduced.

The term bilinear in PLS and PCR does not refer to the relation between the independent variables and the dependent variables but to the decomposition. The meaning of a trilinear decomposition in PLS is that the calibration cube $\underline{\mathbf{X}}$ is decomposed into a set of rank-one cubes describing $\underline{\mathbf{X}}$ in some optimal sense. It does not, however, mean that the part of $\underline{\mathbf{X}}$ relevant for describing the dependent variable $\mathbf{y}$ has to be of rank one. This contrasts with rank annihilation factor analysis[7] or, to some degree, PARAFAC[8] where the contribution of each chemical component is presumed to give rise to a matrix of rank one. By relaxing this constraint, one gains the possibility of being able to cope with matrix effects and intrinsic non-rank-one properties of the analytes. This more or less corresponds to bilinear PLS where the rank of a decomposed matrix can be higher than the number of spectrally active species.

In the literature some attempts have been made to construct a trilinear PLS algorithm.[9,10] These algorithms are very valuable but lack the basic property of PLS, namely maximum covariance between scores and $\mathbf{y}$ in a trilinear sense. The algorithm proposed here has this property and is also very fast. The algorithm proposed by Wold *et al.*[9] is a kind of intermediate between truly trilinear PLS and ordinary bilinear PLS. The cube of calibration matrices is unfolded to give a matrix that contains each sample in one row. This is done by concatenating all rows of a matrix of ones. If the measurements for one sample constitute a $J \times K$ matrix, it becomes instead a JK row vector. A bilinear PLS solution is found for the matrix of all the unfolded sample matrices. In essence this solution is given by the column-wise weight vectors in the $JK \times F$ matrix $\mathbf{W}$, F being the number of components. In a bilinear sense these weight vectors produce scores that have maximum covariance with the unexplained part of $\mathbf{y}$.

The advantages of the proposed tri-PLS are manifold. The main feature of the algorithm is that it produces score vectors that in a trilinear sense have maximum covariance with the unexplained part of the dependent variable. The solution is easy to interpret compared with unfolding methods. This is especially important when the number of variables is high. The algorithm is fast owing to the relatively few parameters to be estimated and the fact that it boils down to eigenvalue problems.

When only a few samples are available, it is important to use the information optimally. Suppose the calibration array is of size $2 \times 10 \times 100$. If the array is unfolded, the result is a 2×1000 matrix. To calculate a 1000-dimensional loading vector, there will then only be two examples. This will certainly not give a very robust estimate of the loading vector. If instead a trilinear decomposition is chosen, the number of examples is significantly elevated. To calculate a ten-dimensional loading vector in the second order, 200 examples are now present, while to calculate a 100-dimensional loading vector in the third order, there are 20 examples. Another obvious advantage of this new calibration method is the possible stabilization of the solution in the case of a low net analyte signal[11-12] because of the incorporation of $\mathbf{y}$ in the decomposition step – a situation where other trilinear models might have difficulties.

There is a price to be paid for a simple model of course. Theoretically speaking, one loses fit in a trilinear model compared with a bilinear model because of the more severe constraints.[13] In practice the problem is not whether fit is lost or not but whether the trilinear model is appropriate or not. This is a problem-dependent question that might not always have a straightforward answer. The applications shown in this paper will illustrate some examples where the trilinear model is appropriate, but in many cases one must try several models to see which ones describe the data best. This would for example be the case if no *a priori* knowledge indicated that a trilinear behaviour was underlying the data. If several models describe the data more or less equally well, one should choose the simplest in order to keep the model robust against overfit and interpretable. Compared with unfolding, the multilinear models are much simpler because they use fewer parameters and are hence preferable with regard to simplicity.

NOMENCLATURE

In the following, scalars are indicated by lowercase italics, vectors by bold lowercase characters, two-way matrices by bold capitals, three-way arrays by underlined bold capitals and four-way arrays by doubly underlined bold capitals. The letters I, J, K, L and M are reserved for indicating the dimension of different orders.

In trilinear PLS each calibration sample is characterized by a $J \times K$ matrix $\mathbf{X}$ of measured variables. This means that for each variable in the order of J the same K variables have been determined (and vice versa). These matrices are collected in a cube $\underline{\mathbf{X}}$ that has the dimension $I \times J \times K$, i.e. there are I samples (first order), J measurements in the second order and K measurements in the third order. For each sample there is a known concentration (or other sought variable or set of variables) to be predicted by the independent variables. These concentrations are collected in an $I \times 1$ vector called $\mathbf{y}$. When there is more than one sought variable, these are collected in a matrix $\mathbf{Y}$ that is of dimension $I \times M$, where M is the number of different analytes.

The general terminology of PLS models is given in Table 1. The prefix indicates the order of $\mathbf{X}$, while the final number indicates the order of $\mathbf{Y}$. As a general term for multiway PLS, N-PLS is proposed.

Preprocessing of three-way tables is a delicate issue that has received some attention in the literature.[14] Centering can be done by first unfolding the calibration array to an $I \times JK$ matrix and then centering this matrix as in ordinary PLS. Scaling in multiway analysis has to be done taking the trilinear model into account. One cannot scale column-wise, but rather whole 'slabs' of the array must be scaled.

Table 1. Abbreviations for different PLS models depending on order of $\mathbf{X}$ and $\mathbf{Y}$

	Order X		
Order Y	2	3	4
1	Bi-PLS1	Tri-PLS1	Quadri-PLS1
2	Bi-PLS2	Tri-PLS2	Quadri-PLS2
3	Bi-PLS3	Tri-PLS3	Quadri-PLS3

THEORY

A trilinear equivalent to bi-PLS1 (one dependent variable) will first be developed. This algorithm is then extended to a trilinear PLS2 algorithm (several dependent variables) and finally it is shown how to make higher-order models, e.g. quadrilinear PLS.

Tri-PLS1

To extend PLS to three-way arrays, we will first elaborate on the essential part of bi-PLS. The ordinary bi-PLS algorithm can be described as essentially consisting of two steps. For each component a model is built of both **X** and **y**, then these models are subtracted from **X** and **y** and a new set of components is found from the residuals. The calculation of components is the main part of the algorithm and will be presented here as a problem of determining a weight vector **w** to maximize a certain function.

To calculate a component in bi-PLS, one seeks a one-component model of **X** of the form

$$\hat{x}_{ij} = t_i w_j \tag{1}$$

where the t_i are called scores and the w_j weights. This model is found by

$$\max_{\mathbf{w}} \left[\operatorname{cov}(\mathbf{t}, \mathbf{y}) \middle| \min \left(\sum_{i=1}^{I} \sum_{j=1}^{J} (x_{ij} - t_i w_j)^2 \right) \wedge \|\mathbf{w}\| = 1 \right] \tag{2}$$

This expression tells us to find a **w** that will yield a least squares solution to the model in equation (1), thereby yielding a score vector **t** with maximal covariance with **y**. When **w** is known and of length one, the least squares solution to the problem of finding **t** is found by projection of **X** on **w**. This can be described in summation signs as

$$\max_{\mathbf{w}} \left(\operatorname{cov}(\mathbf{t}, \mathbf{y}) \middle| t_i = \sum_{j=1}^{J} x_{ij} w_i \wedge \|\mathbf{w}\| = 1 \right) \tag{3}$$

The covariance between **t** and **y** can also be expressed by summations:

$$\max_{\mathbf{w}} \left(\sum_{i=1}^{I} t_i y_i \middle| t_i = \sum_{j=1}^{J} x_{ij} w_j \wedge \|\mathbf{w}\| = 1 \right) \tag{4}$$

This equation is not strictly correct, because there is no correction for degrees of freedom, but since this correction is constant for a given component, it will not affect the maximization. Also, as in ordinary PLS, equation (4) does not truly express the covariance if $\underline{\mathbf{X}}$ has not been properly centered. Instead of the covariance it will then be the uncentered covariation that is maximized. The expressions for covariance and the least squares solution of finding the t_i can be written as

$$\max_{\mathbf{w}} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} y_i x_{ij} w_j \wedge \|\mathbf{w}\| = 1 \right) \tag{5}$$

Since **X** and **y** are known beforehand, the summation over i can be done before actually calculating **w**. This summation will yield a vector of size $J \times 1$ that is called **z**:

$$\max_{\mathbf{w}} \left(\sum_{j=1}^{J} z_j w_j \wedge \|\mathbf{w}\| = 1 \right) \tag{6}$$

Because **w** is restricted to be of length one, the maximal value of the above expression is reached when **w** is a unit vector in the same direction as **z**. Therefore the solution of finding **w** yields

$$\mathbf{w} = \frac{\mathbf{z}}{\|\mathbf{z}\|} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{y}}{\|\mathbf{X}^{\mathrm{T}}\mathbf{y}\|} \tag{7}$$

This in essence explains the bilinear PLS model.

This result is extended to tri-PLS in the following. The goal of the algorithm is to make a decomposition of the cube $\underline{\mathbf{X}}$ into a set of triads. A triad is the trilinear equivalent to a bilinear factor, i.e. a rank-one model of the calibration array. A triad consists of one score vector **t** and two weight vectors, one in the second order called $\mathbf{w}^J$ and one in the third order called $\mathbf{w}^K$. The model of $\underline{\mathbf{X}}$ is given by the equation

$$x_{ijk} = t_i w_j^J w_k^K \tag{8}$$

This equation is shown graphically in Figure 1. By the same reasoning as for bi-PLS we express the tri-PLS model as the problem of finding the vectors $\mathbf{w}^J$ and $\mathbf{w}^K$ that satisfy

$$\max_{\mathbf{w}^J\mathbf{w}^K} \left[\mathrm{cov}(\mathbf{t}, \mathbf{y}) \left| \min\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} (x_{ijk} - t_i w_i^J w_k^K)^2 \right) \right. \right]$$

Both $\mathbf{w}^J$ and $\mathbf{w}^K$ are, as before, of length one but are left out of the expression for simplicity. The least squares solution to the model (8) can be expressed similarly to the bilinear model, giving

$$\max_{\mathbf{w}^J\mathbf{w}^K} \left(\mathrm{cov}(\mathbf{t}, \mathbf{y}) \left| t_i = \sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk} w_j^J w_k^K \right. \right) \tag{9}$$

This implies

$$\max_{\mathbf{w}^J\mathbf{w}^K} \left(\sum_{i=1}^{I} t_i y_i \left| t_i = \sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk} w_j^J w_k^K \right. \right) = \max_{\mathbf{w}^J\mathbf{w}^K} \left(\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} y_i x_{ijk} w_j^J w_k^K \right) = \max_{\mathbf{w}^J\mathbf{w}^K} \left(\sum_{j=1}^{J}\sum_{k=1}^{K} z_{jk} w_j^J w_k^K \right)$$

where **Z** is now a matrix instead of a vector. To maximize this expression, we formulate it in terms of matrices:

$$\max_{\mathbf{w}^J\mathbf{w}^K} [(\mathbf{w}^J)^{\mathrm{T}}\mathbf{Z}\mathbf{w}^K] \Rightarrow (\mathbf{w}^J, \mathbf{w}^K) = \mathrm{SVD}(\mathbf{Z})$$

where SVD(**Z**) means using the first set of normalized vectors from a singular value decomposition (SVD) on **Z**. The problem of finding $\mathbf{w}^J$ and $\mathbf{w}^K$ is simply accomplished by calculating this set of vectors. This follows directly from the properties of these factors/eigenvectors (see e.g. Reference 15, paragraph 3.3.2).

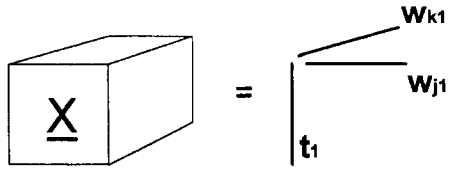

Figure 1. Graphical representation of a one-component trilinear decomposed model of $\underline{\mathbf{X}}$

The complete trilinear PLS1 algorithm is given in Table 2. Since the score vectors from different components are not orthogonal, the calculation of the regression coefficients in step 4 has to be done by taking all calculated score vectors into account. The score matrix $\mathbf{T}$ has the dimension $I \times F$ and contains in the fth column the fth score vector.

If $\underline{\mathbf{X}}$ and $\mathbf{y}$ are not centered, a column of ones must be added to $\mathbf{T}$ to include the offset in the regression.

It is possible to calculate residuals in $\underline{\mathbf{X}}$ by an extra set of loading vectors $\mathbf{p}$ just as in ordinary PLS. To calculate these, steps 1 and 2 in the algorithm are repeated, but now $\mathbf{t}$ takes the place of $\mathbf{y}$, and $\mathbf{p}^J$ and $\mathbf{p}^K$ take the place of $\mathbf{w}^J$ and $\mathbf{w}^K$. However, this will not yield orthogonal scores and is therefore omitted here.

Tri-PLS2

The algorithm outlined above corresponds to bilinear PLS1 in that there is only one dependent variable $\mathbf{y}$. If several dependent variables are present, it is possible to analyze each analyte separately by this algorithm, but one can also calibrate for all analytes simultaneously as in the PLS2 algorithm. The algorithm for several dependent variables, which is iterative, is given in Table 3. The algorithm is seen to be a straightforward extension of the bilinear PLS2 algorithm and to equal trilinear PLS1 if only one dependent variable is modeled.

PLS in higher orders

It is possible to extend the algorithms shown above to any desired order. This will be shown here for quadrilinear PLS1. For calibration purposes we have a vector of e.g. concentrations $\mathbf{y}$ and a

Table 2. Trilinear PLS1

Center $\underline{\mathbf{X}}$ and $\mathbf{y}$
$f = 1$

1. Calculate matrix $\mathbf{Z}$
2. Determine $\mathbf{w}^J$ and $\mathbf{w}^K$ by SVD
3. Calculate $\mathbf{t}$
4. $\mathbf{b} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y}$
5. Each sample $\mathbf{X}_i$ is replaced by $\mathbf{X}_i - t_i\mathbf{w}^J(\mathbf{w}^K)^T$ and $\mathbf{y} = \mathbf{y} - \mathbf{Tb}$
6. $f = f + 1$. Continue from step 1 until proper description of $\mathbf{y}$

Table 3. Trilinear PLS2

Center $\underline{\mathbf{X}}$ and $\mathbf{Y}$
Let $\mathbf{u}$ be equal to a column in $\mathbf{Y}$
$f = 1$

1. Calculate matrix $\mathbf{Z}$ (use $\mathbf{u}$ instead of $\mathbf{y}$)
2. Determine $\mathbf{w}^J$ and $\mathbf{w}^K$ by SVD
3. Calculate $\mathbf{t}$
4. $\mathbf{q} = \mathbf{Y}^T\mathbf{t}/|\mathbf{Y}^T\mathbf{t}|$
5. $\mathbf{u} = \mathbf{Yq}$
7. If convergence, continue; else step 1.
8. $\mathbf{b} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{u}$
9. $\mathbf{X}_i = \mathbf{X}_i - t_i\mathbf{w}^J(\mathbf{w}^K)^T$ and $\mathbf{Y} = \mathbf{Y} - \mathbf{Tbq}^T$
10. $f = f + 1$. Continue from step 1 until proper description of $\mathbf{Y}$

four-way array of measurements $\underline{\underline{\mathbf{X}}}$ ($I \times J \times K \times L$). As shown in the three-way algorithm, a score vector $\mathbf{t}$ is sought with maximum covariation with $\mathbf{y}$. This is obtained by calculating a cube $\underline{\mathbf{Z}}$ where the jklth element is given by the dot-product of $\mathbf{y}$ and the $I \times 1$ column in $\underline{\underline{\mathbf{X}}}$ consisting of all elements with indices (j, k, l).

To find the first component of a quadrilinear PLS solution, we seek the one-component decomposition of this cube that explains most of the variance. This solution can be found by using PARAFAC (which is normally very fast when only one component is sought). The decomposition of the cube by PARAFAC immediately yields three weight vectors $\mathbf{w}^J$, $\mathbf{w}^K$ and $\mathbf{w}^L$ which, normalized, then again determine the score vector. For a five-way table the principle is analogous, but there are now four weight vectors to estimate for each component. These can be found by a quadrilinear PARAFAC one-component model.

These steps show the principle in extending any of the algorithms above to any order. To calculate the one-component PARAFAC models, an algorithm specifically aimed at this purpose has been designed. This algorithm, called N-PCA, is outlined in the Appendix.

It is also possible to extend the outlined PLS algorithm to dependent variables of higher orders. This can be done in a similar way to extending the Xs to higher orders. This is easily seen by realizing that bilinear PLS2 is symmetrical in the way loadings and scores are found for X and Y. Likewise, a trilinear decomposition of $\underline{\mathbf{Y}}$ can be found iteratively by making a matrix corresponding to $\mathbf{Z}$ by properly multiplying $\underline{\mathbf{Y}}$ and $\mathbf{t}$ and then decomposing it by SVD into two loading vectors which then define the score vector $\mathbf{u}$.

ASSESSMENT OF A TRILINEAR PLS SOLUTION

All the qualitative and quantitative tools from bilinear modeling are in principle applicable in the trilinear case for influence and residual analyses (see also Reference 16 for tools specifically targeted at trilinear models).

Leverages can be calculated from the scores or loadings of a given order, giving an indication of the influence/importance of a sample or variable. It should be remembered that neither scores nor loadings are orthogonal, so leverages must be calculated from the general formulae for the Mahalanobis distance.[17]

The degree of overlap between the loadings in one order can give information on the spectral dependences between the latent variables. This might be especially interesting if there is a higher degree of overlap in one order than the other. This could indicate dependency of the species in one direction. In chromatography this could be caused by two species having almost identical retention times. This would degrade the solution and a better separation method would be preferred in this instance.

APPLICATION OF TRILINEAR PLS

There are many possible three-way calibration methods. Among the most common are unfolding and using standard bilinear methods (PCR, PLS, ridge regression), using trilinear PARAFAC (parallel factor analysis) or, to a lesser extent, using modifications of the so-called Tucker models.[16,18] In this paper two illustrative examples of tri-PLS will be given. The results will be compared only with the results of unfold-PLS according to Wold *et al.*,[9] since this method resembles the proposed method to some extent. For a calibration problem with a three-way cube $\underline{\mathbf{X}}$ and one dependent variable $\mathbf{y}$, unfold-PLS is performed by unfolding the three-way cube to a matrix, keeping the order of $\underline{\mathbf{X}}$ in common with $\mathbf{y}$ intact. The matrix formed in this way can be modeled by ordinary bi-PLS.

The goal of this comparison is not to emphasize one method over the other. There is no definitive calibration method that one can stick to. Some general guidelines can be given,[13,19,20] but all methods are to some extent based on certain assumptions regarding the behavior of the data. Since this behavior can vary from data set to data set, so will the choice of appropriate calibration method. The choice of calibration method should always rest upon knowledge and investigation of the data and knowledge about the calibration methods. One can imagine several situations where N-PLS will be an appropriate method.

The advantage of using tri-PLS instead of PARAFAC would be in the speed of model building, since the current PARAFAC algorithms are *very* slow when the number of variables is high. This can be very annoying in practice during cross-validation and outlier detection. Another advantage of tri-PLS over PARAFAC is the incorporation of the dependent variables in the decomposition of the independent variables, which might stabilize the predictive model.

The advantage of using tri-PLS instead of unfolding methods is twofold. The trilinear model is much more parsimonious, i.e. simple, and hence easier to interpret and the trilinear model will also potentially be less prone to noise, because the information across all orders is used for the decomposition.

These advantages of tri-PLS, or N-PLS in general, indicate that there are many possible applications. Calibration/regression is an obvious application, but N-PLS might also be used for explorative data analysis, feature selection and process monitoring. Any N-way data structure can be decomposed, not only multilinear data. In the case of e.g. image analysis the benefit of using the trilinear constraint would be an easier way of finding interesting areas in the images. Though the data may not theoretically be trilinear, the trilinear constraint ensures that the emphasis is on areas (subarrays of variables) instead of single variables. This can greatly simplify and help the interpretation of complex data, where chance correlation or variation can otherwise be a serious obstacle in the search for meaningful models.

To conclude this tribute to N-PLS, some typical ways of achieving multiway data should be mentioned. First of all, obtaining spectra in conjunction with chromatography, kinetics or dynamics data from different locations/occasions/samples/etc. is a straightforward way of getting higher-order data. Using fluorescence, the order of the data can possibly be raised substantially by employing emission, excitation and lifetime. Instead of spectra, other sets of variables can also be used. In psychometrics, typical variables could be a set of scores from several different tests.

MATERIALS AND METHODS

Problem 1. Determination of ash content in sugar by fluorescence spectroscopy

Instrumentation

All experiments were performed on a Perkin-Elmer LS 50B spectrometer. The computer controlling the instrument through an RS232C interface is an IBM-compatible 486 50 MHz PC. Uncorrected fluorescence spectra sampled at 2 nm intervals are recorded in all experiments.

Programs

The algorithms were implemented in Matlab for Windows 4.2c.1 (Mathworks, Inc.) on a DX–4 personal computer. The algorithms are available from World Wide Web:

hhtp:\\newton.foodsci.kvl.dk\foodtech. The Perkin-Elmer LS 50 FLDM instrument program (version 4.00) is used for controlling the instrument.

Samples

Sixty-seven samples of white sugar were obtained from a sugar plant by sampling every second hour. The ash content of the samples was determined by wet chemical methods and the fluorescence measured.

The fluorescence was measured at four excitation (230, 240, 290 and 340 nm) and 63 emission (375–500 nm, 2 nm intervals) wavelengths. The samples were prepared by dissolving 2·25 g sugar in 15 ml buffer. One sample was discarded owing to non-normal appearance.

Doubly ion-exchanged water was used for all sugar solutions and dilutions. There was a high degree of repeatability of the sample measurements. In the experiments, phosphate buffers (pH 7·0) were made from 0·025 M potassium dihydrogen phosphate and 0·025 M sodium hydrogen phosphate adjusted with 1 M NaOH (p.a.). All measurements started with the largest excitation

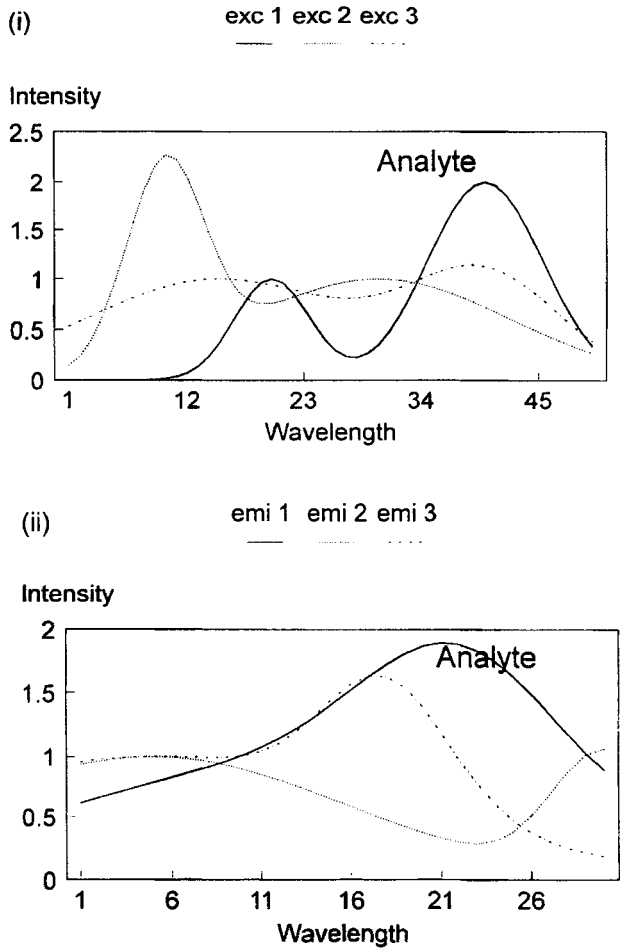

Figure 2. (i) Excitation and (ii) emission spectra used in simulated example

wavelength and ended with the lowest in order to minimize photodecomposition of the sample. In all experiments the sample holder was thermostatted to $20 \pm 0{\cdot}1$ °C, which is the temperature of the thermostatted room.

The ash content of each sample was determined on the sugar plant by a standard EU method (1265/69). The determination is based on a conductivity measurement and is very precise.

Problem 2. Simulated excitation–emission matrices

Simulation

Simulated EEMs were made in Matlab using the excitation and emission spectra shown in Figure 2. Three analytes were present in the samples. The analyte of interest is analyte 1 with excitation spectrum $\mathbf{exc}_1$ and emission spectrum $\mathbf{emi}_1$. The spectra shown correspond to an arbitrary concentration of one. All samples varied independently on five levels of each analyte/interference. The interferences vary in concentration between 1 and 5, whereas the analyte varies between 0·1 and 0·5.

A total of 125 (5^3 design) calibration samples and 125 test samples were generated by summing the concentration-weighted EEM for each analyte. The two arrays of EEMs were each of size $125 \times 50 \times 30$ (sample × excitation × emission). Severe Gaussian noise was added (zero mean, standard deviation 25% of mean signal). Noise was also added to the reference concentrations of the calibration set (approximately 1% of the concentration). No noise was added to the concentrations of the test set, so the calculated prediction errors are the true errors.

RESULTS

Sugar example

The unfolded fluorescence spectra of all samples are shown in Figure 3. The corresponding y-values are shown in Figure 4.

No obvious outliers came up during the analysis, apart from the one that was discarded immediately. The samples were divided into two parts and mean centered. Every second sample

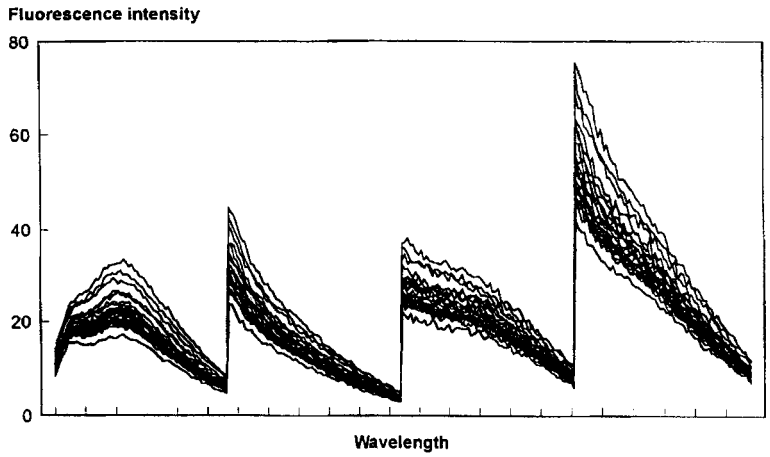

Figure 3. Emission spectra of 63 sugar samples excited at four wavelengths

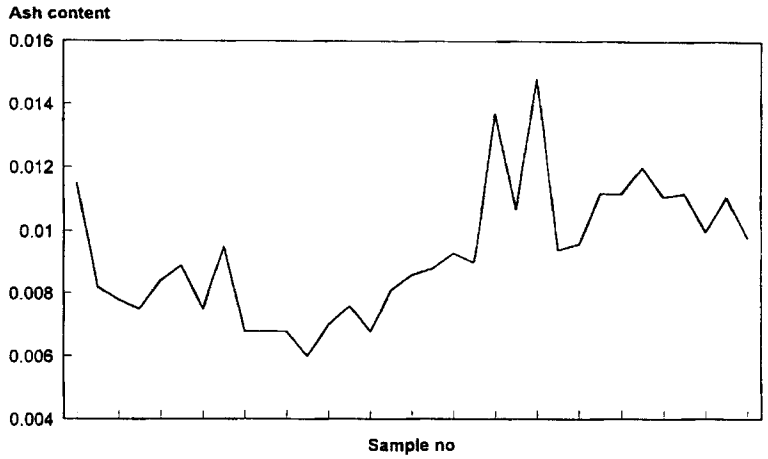

Figure 4. Ash content in sugar samples

constituted the calibration set and the rest a test set. Both unfold-PLS and tri-PLS used two components to give an adequate description of the variations in **y**. The results were almost equal with regard to predictions (Table 4). This is not surprising, since the number of samples is relatively large compared with the number of components. If a trilinear solution is adequate, then a bilinear solution will also in principle be good as long as the noise in $\underline{\mathbf{X}}$ is not too large. When comparing the weight vectors, however, important differences are seen. In Figures 5 and 6 these weight vectors are compared. The stabilization of the solution in the case of a trilinear decomposition is easily seen. The unfold-PLS solution is very difficult to interpret with regard to the last component. In tri-PLS the weight vectors are easier to interpret, because the noise is reduced owing to the fact that fewer parameters need to be estimated.

In unfold-PLS there is a clear difference between the shape of the four parts of the first weight vector corresponding to different excitation wavelengths. unfold-PLS incorporates the differences in the emission spectra in the first dimension, but thereby it is difficult to detect the minor variations, i.e. the fact that the individual emission spectra are present at all excitation wavelengths but in different amounts.

Simulation example

Three components were used in both tri-PLS and unfold-PLS. The results for the independent predictions are given in Table 5. There is a gross difference in the RSD (standard deviation of residuals). This is due to the severe noise on the measured spectra. In unfold-PLS the lack of

Table 4. Results for test set: (i) correlation between concentrations and predictions; (ii) relative standard deviation

	(i) R^2	(ii) RSD
Tri-PLS	0·9396	6·95%
Unfold-PLS	0·9401	7·01%

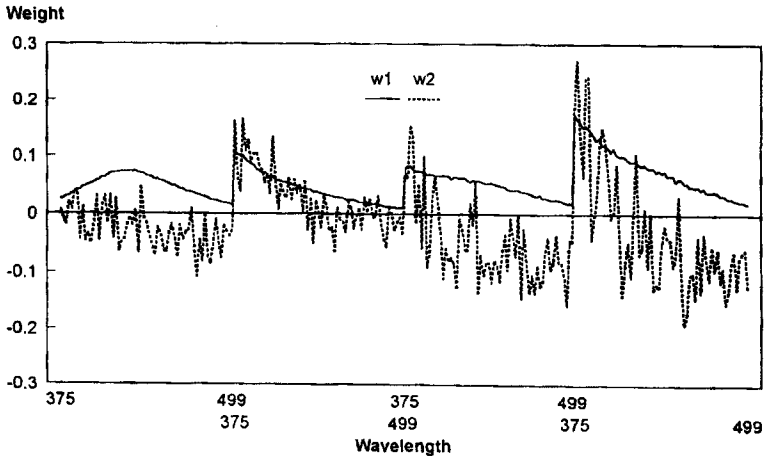

Figure 5. Weight vectors from unfold-PLS

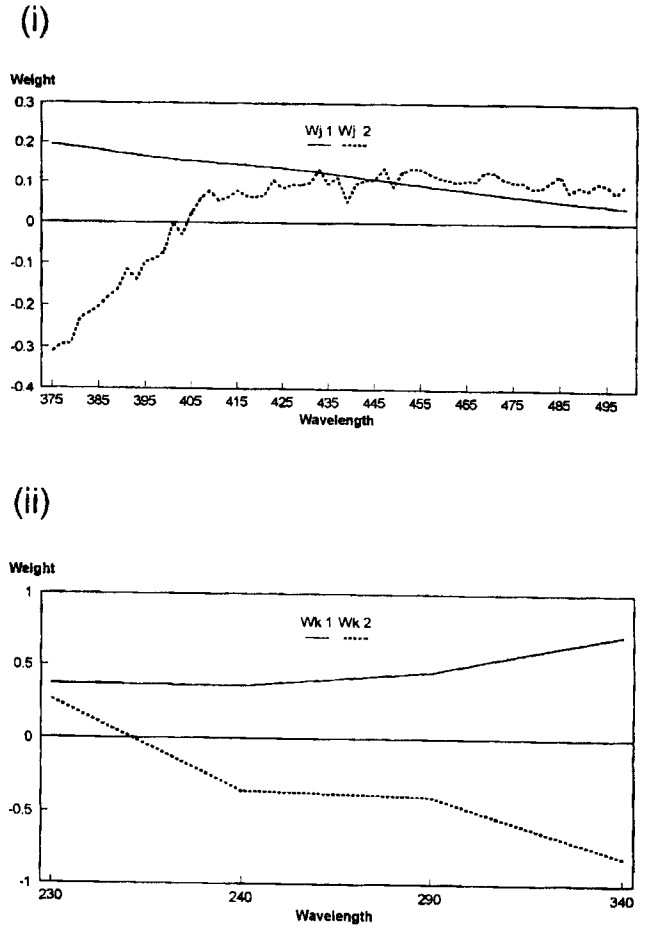

Figure 6. Weight vectors from trilinear PLS in order of excitation (i) and order of emission (ii)

Table 5. Results for test set: (i) correlation between concentrations and predictions; (ii) relative standard deviation

	(i) R^2	(ii) RSD
Tri-PLS	0·9655	12·4%
Unfold-PLS	0·9505	22·3%

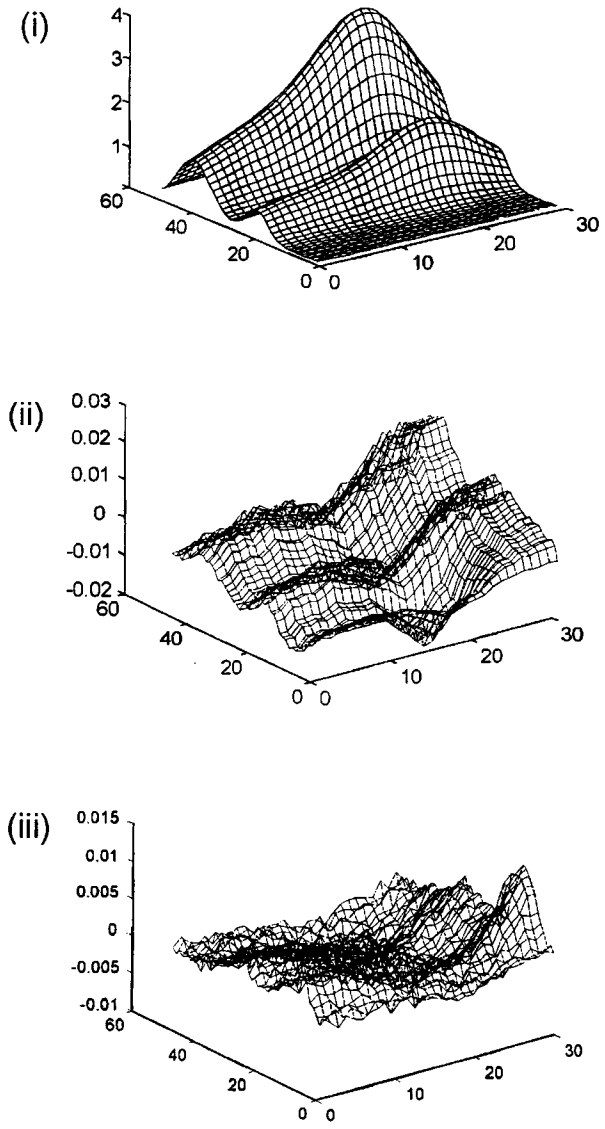

Figure 7. Comparison of true analyte EEM (i) and regression matrix from tri-PLS (ii) and unfold-PLS (iii)

the trilinear constraint has the consequence that the information across excitation wavelengths is not used to stabilize the solution. The result of this is that the model fits more noise than does tri-PLS. This can also be seen by the fact that the RSD for tri-PLS is more or less the same for the calibration and the test set ($RSD_{pred} = 12{\cdot}4\%$ and $RSD_{cal} = 10{\cdot}8\%$ respectively), whereas for unfold-PLS the picture is quite different ($RSD_{pred} = 22{\cdot}3\%$ and $RSD_{cal} = 8{\cdot}0\%$). To see how the increased stability influences the interpretability of the solution, one can compare the regression matrix (corresponding to the regression vector in bi-PLS) with the true underlying EEM for the analyte (Figure 7). Though the noisy data degrade both solutions, it is obvious that tri-PLS is less prone to being influenced by the noise. The regression matrix for tri-PLS is more structured (some resemblance to the true EEM for the analyte), which is not the case for unfold-PLS.

CONCLUSIONS

The N-PLS model proposed, and exemplified with trilinear applications, is a natural extension of the bilinear PLS model. The strength of the method is its stabilized solution and interpretability compared with unfold-PLS. Another important aspect is the speed of the algorithm, which is especially important when the number of variables is very high.

Further investigation will be aimed at comparing tri-PLS more thoroughly with several calibration methods based on multilinear decompositions, e.g. different methods based on PARAFAC, to determine whether the simultaneous decomposition performed in PARAFAC has an influence on the predictive ability.

ACKNOWLEDGEMENTS

I am grateful for support and inspiration from and funds to Professor Lars Munck from Nordic Industry Foundation Project P93149 and the FØTEK Foundation. Age K. Smilde, Henk Kiers and Lars Nørgaard are thanked for invaluable comments and hints.

APPENDIX: TRILINEAR PCA

It is possible to modify the above algorithms to a trilinear PCA algorithm which calculates a triad that describes the maximal variance of an array of variables. For a one-component solution this equals a PARAFAC solution.

For a three-way array the first set of components in $\underline{\mathbf{X}}$ is sought to explain the maximum variance in $\underline{\mathbf{X}}$. The trick is simply to replace $\mathbf{y}$ in the PLS1 algorithm by $\mathbf{t}$. Then, however, since $\mathbf{t}$ is not given in advance, the algorithm becomes iterative. First, $\mathbf{t}$ is presumed known and $\mathbf{Z}$ is calculated as

$$z_{jk} = \sum_{i=1}^{I} t_i x_{ijk} \tag{10}$$

i.e. with $\mathbf{y}$ replaced by $\mathbf{t}$. Then estimates of $\mathbf{w}^J$ and $\mathbf{w}^K$ are determined by SVD. Afterwards $\mathbf{w}^J$ is assumed known and from the symmetry of the problem $\mathbf{w}^J$ can replace $\mathbf{y}$ instead of $\mathbf{t}$, and $\mathbf{t}$ and $\mathbf{w}^K$ can be estimated. This procedure is continued until convergence. The algorithm can be written as in Table 6, where it is generalized to the calculation of several components, though only one component is used in the PLS algorithm. As a starting guess for $\mathbf{t}$ according to step 1, one can use the first score vector from an SVD of an unfolded cube $\underline{\mathbf{X}}$ (keeping the first order intact) to speed up the algorithm. When calculating the residual of $\underline{\mathbf{X}}$, it should be remembered

Table 6. Trilinear PCA

Center $\underline{\mathbf{X}}$
$f = 1$
1. Determine start value for $\mathbf{t}$ (see below)
2. Calculate matrix $\mathbf{Z}$ (use $\mathbf{t}$ instead of $\mathbf{y}$)
3. Determine $\mathbf{w}^J$ and $\mathbf{w}^K$ by SVD
4. Fix $\mathbf{w}^J$ and calculate $\mathbf{t}$ and $\mathbf{w}^K$ by steps 2 and 3 with $\mathbf{w}^J$ and $\mathbf{t}$ replacing each other
5. Do the same as step 4 but fix $\mathbf{w}^K$
6. Repeat from step 2 until convergence
7. Let $\underline{\mathbf{X}}$ equal unexplained part of $\underline{\mathbf{X}}$ $(\mathbf{X}_i - t_i \mathbf{w}^J (\mathbf{w}^K)^T)$
8. $f = f + 1$. Continue from step 1 until proper description of $\underline{\mathbf{X}}$

that the vectors of the triad are of norm one. To calculate the complete model, one has to include the singular value (it can be incorporated into $\mathbf{t}$).

It is seen that two of the vectors comprising a triad are calculated given the third. This is done iteratively until convergence. The result is a triad with the property of describing the maximum variance in $\underline{\mathbf{X}}$ under the constraint of trilinearity. This model can also be used for qualitative interpretation or for regression, just as in bilinear PCA. The algorithm usually converges in very few steps (less than 10) and is easily extendable to higher orders.

REFERENCES

1. O. M. Kvalheim, *Chemometrics Intell. Lab. Syst.* **2**, 283–290 (1987).
2. A. Höskuldsson, *J. Chemometrics*, **2**, 211–228 (1988).
3. E. Sanchez and B. R. Kowalski, *J. Chemometrics*, **2**, 247–263 (1988).
4. I. S. Helland, *Commun. Stat. Simul. Comput.* **17**, 581–6107 (1988).
5. P. M. Lang and J. H. Kalivas, *J. Chemometrics*, **7**, 153–164 (1993).
6. G. Kateman, *Chemometrics Intell. Lab. Syst.* **19**, 135–142 (1993).
7. L. Nørgaard and C. Ridder, *Chemometrics Intell. Lab. Syst.* **23**, 107–114 (1994).
8. R. A. Harshman and M. E. Lundy, *Comput. Stat. Data Anal.* **18**, 39–72 (1994).
9. S. Wold, P. Geladi, K. Esbensen and J. Øhman, *J. Chemometrics*, **1**, 41–56 (1987).
10. L. Ståhle, *Chemometrics Intell. Lab. Syst.* **7**, 95–100 (1989).
11. A. Lorber and B. R. Kowalski, *J. Chemometrics*, **2**, 67–79 (1988).
12. M. B. Seasholz and B. R. Kowalski, *Appl. Spectrosc.* **44**, 1337–1348 (1990).
13. H. A. L. Kiers, *Psychometrica*, **56**, 449–470 (1991).
14. A. K. Smilde, *Chemometrics Intell. Lab. Syst.* **5**, 143–157 (1992).
15. E. R. Malinowski, *Factor Analysis in Chemistry*, Wiley, New York, (1991).
16. A. K. Smilde and D. A. Doornbos, *J. Chemometrics*, **6**, 11 –28 (1992).
17. S. Weisberg, *Applied Linear Regression*, Wiley, New York (1985).
18. A. K. Smilde, Y. Wang and B. R. Kowalski, *J. Chemometrics*, **8**, 21–36 (1993).
19. M. B. Seasholz and B. R. Kowalski, *Anal. Chim. Acta*, **277**, 165–177 (1993).
20. E. V. Thomas and P. D. Haaland, *Anal. Chem.* **62**, 1091–1099 (1990).