Journal of Pharmaceutical & Biomedical Analysis
Vol. 9, No. 8, pp. 671–678, 1991
Printed in Great Britain

0731–7085/91 $3.00 + 0.00

Relating multivariate time-series by linear three-way decomposition (LTD) and partial least squares (PLS) analysis*

LARS STÅHLE

Department of Pharmacology, Karolinska Institute, Box 60 400, S-104 01 Stockholm, Sweden

Abstract: The problem of relating multivariate time-series which is common in drug development is considered. The mathematical and statistical problems involve relating two three-way tables. These tables have objects and time-points in common while the variables to be related are unique for each table. A modification is presented of the linear three-way decomposition (LTD) algorithm which directly incorporates the information that both objects and time-points are common to the two tables. A comparison is made with partial least squares (PLS) analysis both at the theoretical level and in their application to three sets of real data. Limitations of LTD are discussed, in particular the constraint imposed by the trilinearity requirement, and areas for future development are proposed.

Keywords: *Multivariate statistics; partial least squares; time-series.*

Introduction

Analysis of multivariate time-series can be carried out by relating two three-way tables in which objects and time-points are common to the two tables while the variables to be related are unique to each table (Fig. 1). This kind of problem is common in many areas of research, notably in drug development-related sciences such as analytical chemistry and pharmacology. The examples in the following text are taken from the author's work in pharmacology.

A major line of development in chemometrics is that of analysis of multi-way (-mode) tables (arrays, matrices) (Fig. 1). These developments, and applications in the psychometrics literature, are reviewed in considerable detail elsewhere [1]. The partial least squares (PLS) approach is the method with the widest general application [2, 3]. Recently, an alternative algorithm was suggested for linear three-way decomposition (LTD) of two related three-way tables [4]. This method is based on tensor algebra, which is relatively simple to understand and aids understanding how the method works. Wold (personal communication, lectures given at MULDAST 1988) uses a scheme of criss-cross projections of matrices onto vectors to explain the way PLS works. This scheme in combination with tensor algebra can be used to modify LTD in a way suggested below. The impetus for this development is that models in which several modes are in common between the matrices to be related (say x and y) are frequently encountered. An example is when a number of objects (mode 1) are observed at a number of time points (mode 2) and a number of variables measured at each time point (mode 3) are related to other variables measured at the same time points (mode 4). Thus x consists of modes 1, 2 and 3 and y is built up from modes 1, 2 and 4.

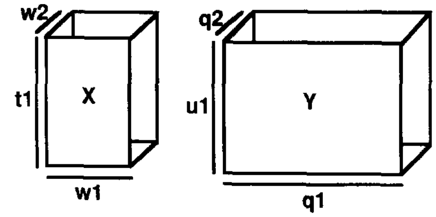

Figure 1
Illustration of the tables (arrays) used in LTD. Notice in particular that t and u are in modes common to x and y and that the same holds true for w_2 and q_2 while w_1 is in a mode unique to x and q_1 is in a mode unique to y.

The aim of the present paper is to compare methods in a natural way that incorporates the information contained in the fact that both

*Presented at the Symposium on "Chemometrics in Pharmaceutical and Biomedical Analysis", November 1990, Stockholm, Sweden.

objects and time-points are common to two related tables. One method is a modification of the LTD method previously developed for the situation in which two modes are unique to each table [4]. The modified LTD method will be referred to as LTD2 (2 modes in common between x and y) and the previous model as LTD1 (1 mode common to x and y). The other method is PLS [2, 3].

Methods

Mathematical model

The detailed description of the LTD2 model and algorithm is given in Appendix A and PLS is described in Appendix B. In short, LTD2 uses data organised into two three-way tables (three-array) in which objects (patients, experimental units) and time-points are in common for the two tables. The result of the analysis is a number for each object describing how much "effect" there is on the object compared to the overall mean. For each time-point a number is given as a measure (weight) of how much effect there is at that time-point. The value of the weight varies in the interval [−1.0; 1.0] in which extreme values indicate a strong influence and values close to 0.0 indicate no influence. Similarly a weight is given for each variable to describe the influence of that variable. To give an example from the Results section on coagulation data: the large negative weight of coagulation factor X and a large positive weight on day 2 for the predictors and the large negative weight for the prothrombin assay data and large positive weights on days 4 and 5 should be interpreted in the following way. Low factor X values on day 2 predict low prothrombin levels on days 4 and 5.

Data

Three sets of real data were used in this initial test of the algorithm. In two of them there are three predictor variables, namely dopamine, DOPAC and 5HIAA and one predicted variable, HVA. All variables were measured in samples taken from the extracellular fluid of the rat brain as microdialysis samples. In the first set of data, six samples were taken in 20 min fractions after injection of amphetamine 5 mg kg^{-1} (n = 5) or saline (n = 5). In the second data set 10 samples of the same kind were taken after administration of α-methyl-p-tyrosine (αMPT) 50 mg kg^{-1} (n = 7), 100 mg kg^{-1} (n = 7), 200 mg kg^{-1} (n = 6) or saline (n = 6). The third data set, which was collected from patients treated with coumarin derivatives after venous thrombosis, consisted of measurements of three coagulation factors (II, VII and X) and five commercially available prothrombin assays (abbreviated SPA, NYC, SIA, TPC and TPR). Samples were collected before treatment and 2, 4 and 5 days after the start of the treatment. The number of patients was 16.

Amphetamine data have the x-array as a (10 × 3 × 6) array and y is (10 × 1 × 6), αMPT data are x (26 × 3 × 10) and y (26 × 1 × 10) and finally coagulation data x (16 × 3 × 4) and y (16 × 5 × 4). All data were normalized to zero mean and unit variance with respect to the variables. Cross-validation was performed in four rounds (see Ståhle, 1989 [4] for details). All data sets contained a small number of missing values.

Results

General observations

The LTD2-algorithm was slow to converge but did so reliably during cross-validation for all significant components (the significance was judged by comparison with unfolded PLS-models). A problem encountered was that the sign of parameters may switch from one iteration to the next for which allowance has to be made. All data were close to being trilinear. This follows from the finding that there was little improvement in the cross-validation estimate of the prediction error obtained by using a PLS-model (unfolded).

Amphetamine data

A scatter plot of the first two components is shown in Fig. 2. From this graph it is clear that amphetamine-treated rats differ from the saline-treated controls. A plot of the weights (Fig. 2b and c) of the first component shows that increased levels of dopamine and decreased levels of DOPAC are strongly correlated to reduced levels of HVA. 5HIAA does not have this effect. The better predictor is DOPAC and all time-points are informative.

αMPT data

A scatter plot of the first two components is shown in Fig. 3(a). On the first component there is a clear separation of saline-treated controls from αMPT-treated animals. The weights (Fig. 3b) show that a reduction of

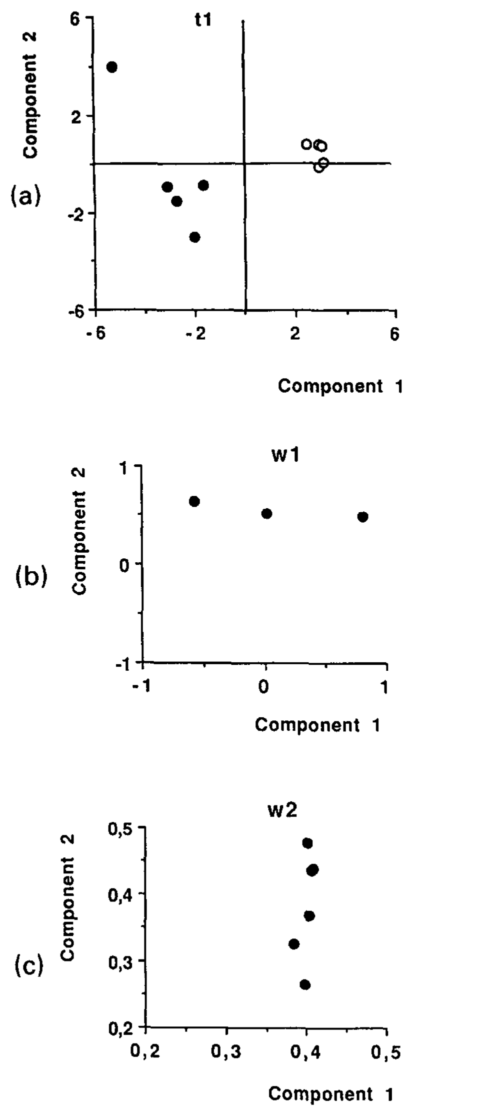

Figure 2
Bivariate plots of the first two dimensions of t_1 (○, saline treated controls; ●, amphetamine treated rats), w_1 and w_2 of the amphetamine data.

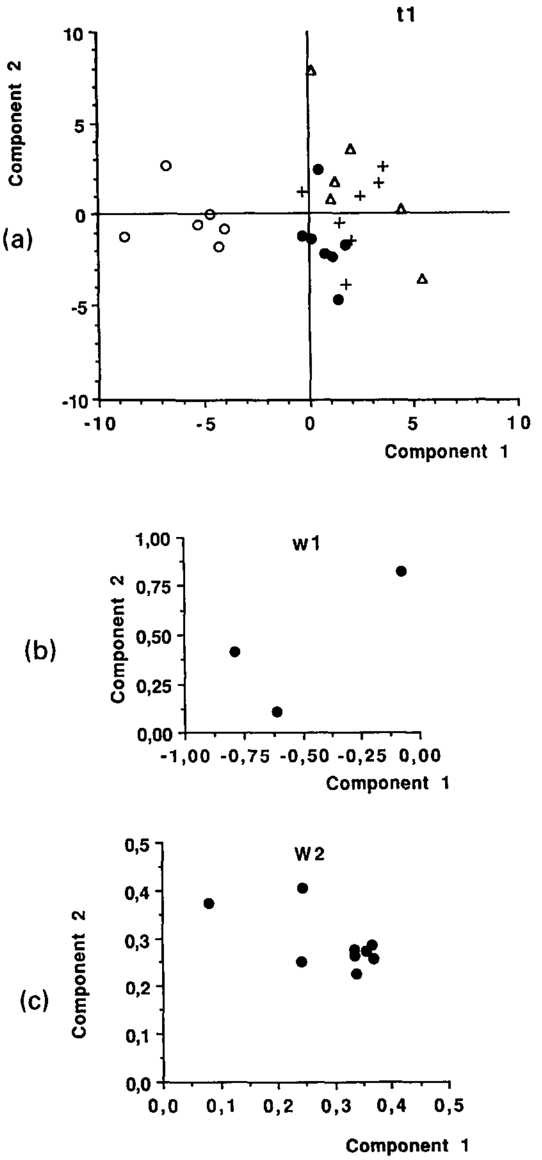

Figure 3
Bivariate plots of the first two dimensions of t_1 (○, control rats; ●, 50 mg kg^{-1}, +−100 mg kg^{-1}, Δ-200 mg kg^{-1}), w_1 and w_2 of the α-methyl-p-tyrosine data.

dopamine and DOPAC is accompanied by a reduction of HVA while 5HIAA did not influence HVA. The influence of dopamine and DOPAC on HVA develops during the first four (Fig. 3c). A second component was also significant and shows the influence of 5HIAA and DOPAC on HVA during the first hour in particular (Fig. 3b and c).

Coagulation data

Coagulation data was the only data set investigated with three-way tables in both x and y. Only one component was found to be significant by cross-validation data. The scores on the first two components are shown in Fig. 4(a). The weights (Fig. 4b,c) show that factor X was most strongly related to the prothrombin assay data (Fig. 4d,e) and that early changes in coagulation factors were related to changes in prothrombin levels on the fourth and fifth days after the start of the treatment. Interestingly, one of the prothrombin assays, TPC, was not as strongly related to the coagulation factors as the other assays.

Comparison with bilinear and three-way PLS-models

The comparison between PLS and LTD2 was made on a cross-validation prediction

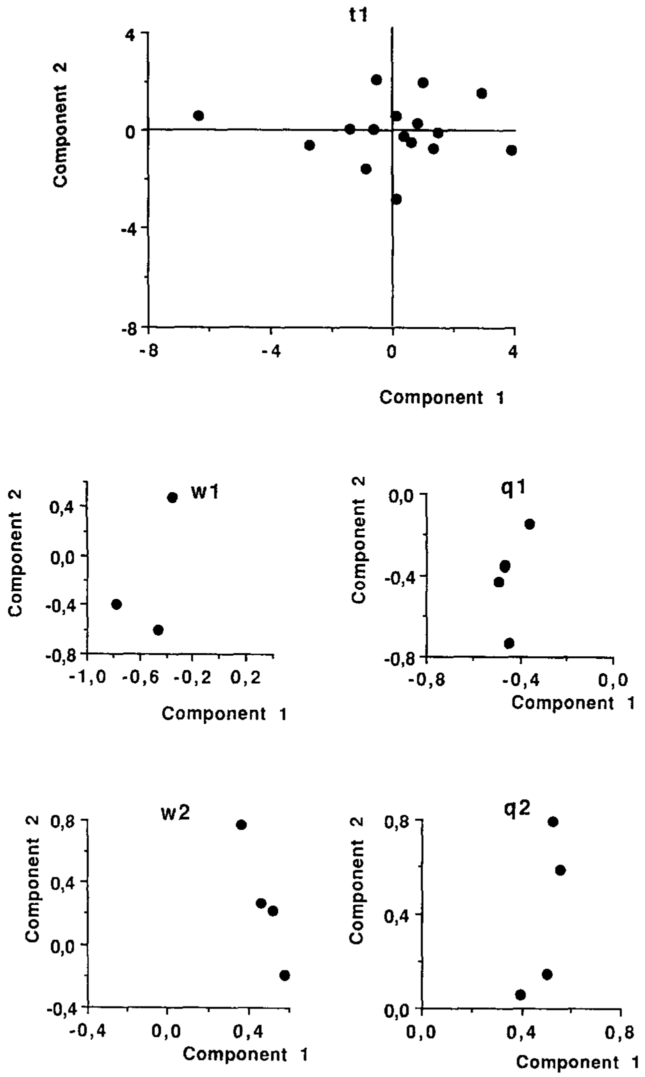

Figure 4
Bivariate plots of the first two dimensions of t_1, w_1, w_2, q_1 and q_2 of the coagulation factors vs prothrombin assay data.

error basis (Table 1). A direct comparison between three different PLS-models and LTD2 was made. It should be noted that the three-way PLS model with one-component PCA decomposition of the weight matrix (abbreviated 3PLS1c) is identical with LTD1 as shown in Appendix B. It is noteworthy that the differences obtained are not large except that the 3PLS2c model (as 3PLS1c except that two PCA components are calculated from the weight matrix) of the coagulation data is superior with respect to the second component indicating that this data set is not well approximated by a trilinear model.

Table 1
Comparison between cross-validation standard deviation ratio for models with one or two components using LTD2, bilinear PLS (PLS), rank 1 (3PLS1c) and rank 2 (3PLS2c) decomposed weight table using multiway PLS

	LTD2	PLS	3PLS1c	3PLS2c
Amphetamine				
1st comp	0.4620	0.4713	0.4624	0.4712
2nd comp	0.7731	0.7896	0.7573	0.7909
αMPT				
1st comp	0.6787	0.6940	0.6865	0.6935
2nd comp	0.9304	0.9255	0.9363	0.9265
Coagulation				
1st comp	0.7399	0.7263	0.7381	0.7320
2nd comp	0.9964	0.9830	0.9818	0.9661

Discussion

The main aim of the present work was to develop an algorithm which takes the constraints inherent in multivariate time-series into account when the model parameters are calculated. Apparently LTD2 does this successfully as shown by the results presented. The more intruiging question is to what extent this approach is successful in handling real data. This prompted a comparison with PLS.

Although computationally demanding, LTD2 produces results that easily lend themselves to interpretations that can be stated in a few sentences. This is a consequence of the trilinearity of the models. The best of the present examples is probably the coagulation data set which show clearly that changes in all coagulation factors, but in particular that of factor *X*, predict reductions in prothrombin levels a few days later (Fig. 5a and b). This interpretation is supported by clinical data which show that the effect of coumarin-treatment is to stop synthesis of coagulation factors in the liver. Due to the half-life of the coagulation factors (i.e. factor II, 50 h; factor VII, 6 h; and factor X, 36 h), the decline in prothrombin-activity a few days after treatment is expected.

Another advantage is that a constraint that is physically present in the data is, in a natural way, incorporated into the model. It would not be surprising if this property results in improved prediction properties compared to LTD1 (3PLS1c). Table 1 is compatible with this conclusion although the improvement is small and confined to the first component. Reasons for this and the trilinearity constraint are discussed below.

The main disadvantage of LTD2 is the risk of trying to model data that are not trilinear. The second component of the coagulation data may serve as an example. Interestingly, the bilinear model (PLS) was not better than the 3PLS2c model which underlines the conclusion previously drawn that real data with physical constraints usually lie between trilinear and bilinear models [4].

It is also interesting to note that, despite considerable efforts to improve the PLS-algorithm in various ways such as linear constraints [3, 4, present model], non-linearity in the relation between *x* and *y* [5] or modification of the parameter estimation procedure (Ståhle; lecture given at Chemometrics 15-year anniversary at Holmsund 1989, to be published), only marginal improvements have been made. This observation suggests that the ordinary PLS algorithm [2] has inherent properties that make it suitable for an unusually large number of problems. Its only serious disadvantage is the interpretation or explanation of the data which may require the expertise of data analysts within applied sciences such as chemometrics to provide knowledge in matrix algebra.

As a final comment, the author proposes the use of methods like those described in this text in the development of new drugs and other areas of pharmaceutical research. The methods are not difficult to use but users require

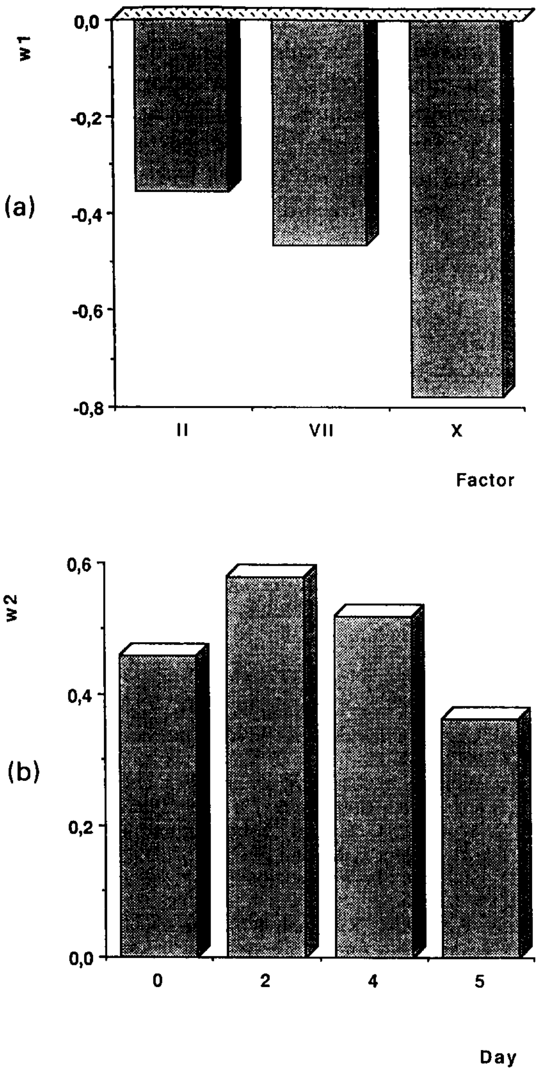

Figure 5
Bar graphs of the first dimension of w_1 (coagulation factors) and w_2 (time points). For interpretations, see the text.

training in precisely the same way as those of any analytical technique and of pharmacological model systems. In addition, the use of multivariate methods gives the user an indication of the limitations of the experimental methods and this may prove to be invaluable. For example the author has observed the situation in which a year was spent in an attempt to optimize an HPLC method in a way which is easily demonstrated to be impossible to achieve. This is mentioned not to scare but to encourage! The present paper provides a way to avoid nasty pitfalls of this type.

Acknowledgements — The present study was supported by the Karolinska Institute, the Swedish Medical Research Council grant no. 09069 and the Svenska Läkarsällskapet. Data from studies in collaboration with Prof. Urban Ungerstedt and Prof. Nils Egberg were used for illustration. The excellent artwork was provided by laboratory engineer Ewa Ljungdahl Ståhle.

References

[1] P. Geladi, *Chemometrics Intell. Lab. Systems* **7**, 11–30 (1989).

[2] S. Wold, A. Ruhe, H. Wold and W.J. Dunn III, *SIAM J. Sci. Statist. Comput.* **5**, 735–743 (1984).

[3] S. Wold, P. Geladi, K. Esbensen and J. Öhman, *J. Chemometrics* **1**, 41–56 (1987).

[4] L. Ståhle, *Chemometrics Intell. Lab. Systems* **7**, 95–100 (1989).

[5] S. Wold, N. Kettaneh-Wold and B. Skagerberg, *Chemometrics Intell. Lab. Systems* **7**, 53–65 (1989).

[6] A. Höskuldsson, *J. Chemometrics* **2**, 211–228 (1988).

[*Received for review 26 November 1990; revised manuscript received 21 January 1991*]

Appendix A

Mathematics

All data will be considered as arrays with specified modes. Thus, a 0-array has no mode and is the same as a scalar. A 1-array has one mode which is specified by an integer and represents a vector. Matrices have two integer-specified modes and are 2-arrays.

In addition to the usual summation formulae and matrix algebra the following operations are defined.

(1) Expansion — symbolized by $\otimes$ which is the same as Kronecker product in the case of two 1-arrays. Thus the expansion of the 1-arrays x and y with modes (1) and (3), respectively yields a 2-array z of mode (1, 3). This is written $z = x \otimes y$ (Fig. A1). It follows that arrays with modes in common cannot be expanded.

(2) Contraction — symbolized by $\odot$ is summation of paired products of elements of modes in common between the arrays contracted. This is the same as scalar product in the case of two 1-arrays. Thus, the contraction of a 3-array x with mode (1, 3, 4) with a 1-array y of mode (3) yields a 2-array z of mode (1, 4). This is written as $z = x \odot y$ (Fig. A1). Orthogonality is defined as the case when the contraction of two arrays of identical mode is zero. The norm of an array, written $\|x\|$, is the contraction of the array with itself, thus $\|x\| = x \odot x$.

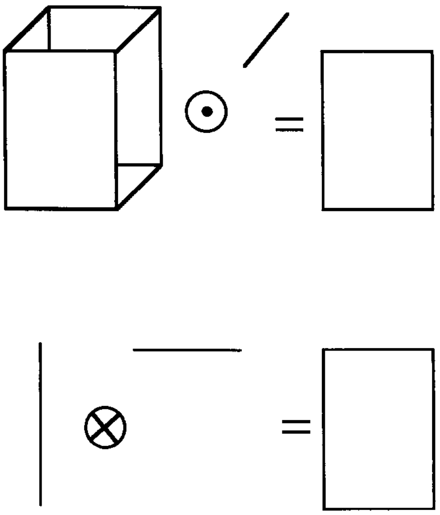

Figure A1
Illustration of the contraction and expansion operations.

The modes of various arrays are always implicit. The reason for introducing this notation is to avoid excessive use of indices and various complicated transposition procedures in connection with array multiplication. It should be noted that the notation is not sufficient to cover matrix algebra in general; the purpose is to simplify and make transparent the decomposition of 3-arrays into 1-arrays. This is possible in the present context only because modes are distinct and correspond to a physical reality.

The following arithmetic properties are of importance: let x and y be arrays with all modes in common,

$$x \odot y = y \odot x. \tag{1}$$

Let x and y have no modes in common. Also let z have modes in common with both x and y,

$$x \otimes y = y \otimes x \tag{2}$$

$$z \odot (x \otimes y) = (z \odot x) \odot y. \tag{3}$$

Notation and models

The predictor array is denoted x and the predicted array is denoted y. The basic idea is to extend the way modes are distinguished in PLS. In PLS a distinction is made between objects (common to x and y) and variables (unique to x and unique to y). In LTD2 it is necessary to distinguish among modes in common between x and y, modes unique to x and modes unique to y. In the following the modes have the following meaning: mode 1 = objects; mode 2 = variables in x; mode 3 = time; mode 4 = variables in y. In parallel with PLS notation there are the following arrays to model/predict x and y.

t_1 Is the score 1-array of mode (1) for x. Mode (1) is common to x and y and is objects (rats, patients, drugs, etc.).

The 1-array u_1 has the same definition for y as t_1 has for x. The mode of t_1 and u_1 is the same.

The 1-array w_2 and w_3 are the weight vectors of modes (2) and (3) for the predictor block x. Two w-arrays are needed, one with mode in common with y (time) and one unique to x (variables).

The corresponding two weight 1-arrays for y are denoted q_4 and q_3, one of which has mode in common with x (q_3) and one has mode unique to y (q_4).

The 2-array of loadings p_{23} is used to calculate residuals of x and is merely a means to obtain orthogonality. It has mode (2, 3).

The scalar d is the regression coefficient between x and y.

For the purpose of writing the iterative algorithm it is necessary to distinguish between arrays extracted in the previous iteration from new arrays by underlining the former.

Properties designed in LTD2

A number of properties useful in data analysis can be found in PLS, in particular orthogonality between certain vectors [6]. The most important is orthogonality between t-vectors of successive components which is also designed into LTD2.

LTD2 algorithm

```
0. Take starting guesses of $t_1$, $u_1$, $q_3$ and $w_3$
1. $\|w_3\| = 1$
2. $\|q_3\| = 1$
3. $w_2 = x \odot (u_1\ q_3)/\|u_1 \otimes q_3\|$
4. $\|w_2\| = 1$
5. $w_3 = x \odot (t_1 \otimes w_2)/\|t_1 \otimes w_2\|$
6. $t_1 = x \odot (w_2 \otimes w_3)/\|w_2 \otimes w_3\|$
7. $\|w_2 \otimes w_3\| = 1$
8. $q_4 = y \odot (t_1 \otimes w_3)/\|t_1 \otimes w_3\|$
9. $\|q_4\| = 1$
10. $q_3 = y \odot (u_1 \otimes q_4)/\|u_1 \otimes q_4\|$
11. $u_1 = y \odot (q_3 \otimes q_4)/\|q_3 \otimes q_4\|$
12. $\|q_3 \otimes q_4\| = 1$
13. Check for convergence of the w- and u-arrays. If not, go to step 1
14. $p_{23} = x \odot t_1/\|t_1\|$
15. $\varphi = \|p_{23}\|$
16. $p_{23} = p_{23}/\varphi$
17. $t_1 = t_1 \cdot \varphi$
18. $w_3 = w_3/\varphi$
19. $d = t_1 \odot u_1/\|t_1\|$
20. $e = x - t_1 \otimes p_{23}$
21. $x = e$
22. $f = y - d \cdot (t_1 \otimes w_3 \otimes q_1)$
23. If you want the next component go to 1
```

The main difference compared to PLS and LTD1 algorithm is that the constraint is made from the other block.

Appendix B

In this appendix the steps in the PLS algorithm and the LTD1 algorithm of importance for the proof given in appendix C are reviewed. Complete descriptions of the algorithms can be found elsewhere [2–4]. The score and weight vectors are identical in notation with LTD2. However, the algorithms are here given as summation formulae and therefore indices corresponding to the modes are used. Hence, the indices are: mode (1) = i, mode (2) = j or J (unfolded), mode (3) = k (now unique to x), mode (4) = l or L (unfolded), mode (5) = m (a new mode unique to y).

```
0. Guess $u_i$
1. $w_J = \sum_i x_{iJ} u_i / \sum_i u_i^2$
2. $\sum_J w_J^2 = 1$
3. $t_i = \sum_J x_{iJ} w_J / \sum_J w_J^2$
4. $q_L = \sum_i y_{iL} t_i / \sum_i t_i^2$
5. $\sum_L q_L^2 = 1$
6. $u_i = \sum_i y_{iL} q_L / \sum_L q_L^2$
7. Repeat steps 1–6 until convergence of $u$
```

In the case of three-way PLS the vector w may be folded (Fig. A2) to make up a matrix W with rows corresponding to time-points and columns corresponding to variables (in the present case). The matrix W can then be decomposed by principal components analysis (PCA) into a number of components (determined by cross-validation on Y [4]). The principal component model of W is then unfolded to make up the vector w in step 3. With a one-component PCA model of W the complete PLS model is referred to as 3PLS1c, with a two-component PCA model of W it is 3PLS2c, etc. Formally the algorithm is written (with score τ and weights π of W):

```
2a. Guess $\tau_j$
2b. $\pi_k = \sum_j W_{jk} \tau_j$
2c. $\sum_k \pi_k^2 = 1$
2d. $\tau_j = \sum_k W_{jk} \pi_k$
```

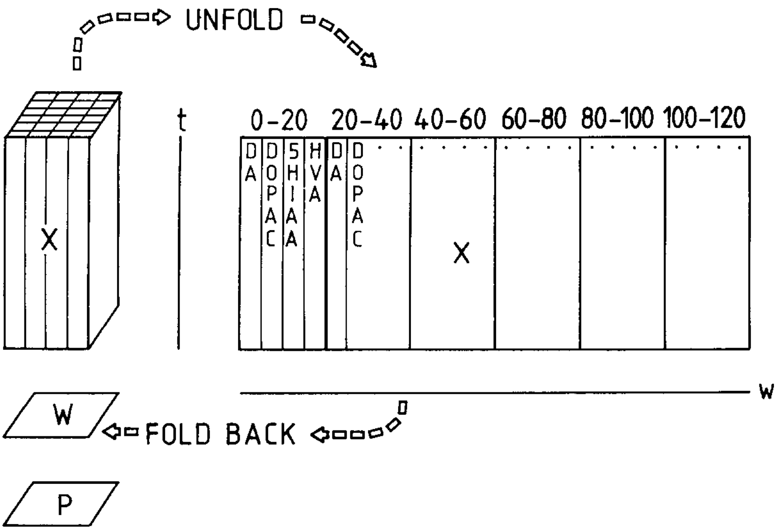

Figure A2
Illustration of the folding and unfolding of x and w in multiway PLS.

2e. Repeat steps 2b–d until convergence, unfold W_{jk} to the vector with elements w_J. The same procedure can be used on q in steps 4a–e exactly corresponding to 2a–e.

The LTD algorithm decomposes the three-way matrix x into three vectors t_1 (same as in PLS), w_2 and w_3 (the product $w_2 \otimes w_3$ yields a matrix the elements of which are the same as the elements in the vector w folded to W in PLS). Indices are used as described above.

1. Make starting guesses for u_1, w_2, w_3, q_4 and q_5
2. $w_j = \sum_i \sum_k x_{ijk} u_i w_k$
3. $\sum_j w_j^2 = 1$
4. $w_k = \sum_i \sum_j x_{ijk} u_i w_j$
5. $\sum_k w_k^2 = 1$
6. $t_i = \sum_j \sum_k x_{ijk} w_j w_k$
7. $q_l = \sum_i \sum_m y_{ilm} t_i q_m$
8. $\sum_l q_l^2 = 1$
9. $q_m = \sum_i \sum_l y_{ilm} t_i q_l$
10. $\sum_m q_m^2 = 1$
11. $u_i = \sum_l \sum_m y_{ilm} q_l q_m$
12. Repeat steps 2–11 until convergence.

Appendix C

It is proposed that LTD1 is identical with PLS (more exactly 3PLS1c) with a one-dimensional principal components (singular value) decomposition of W in the algorithm (not afterwards). It will suffice to show that at convergence the calculation of t_1, u_1, q_4 and q_5 (modelled as the matrix Q by $q_4 \otimes q_5$ in LTD1) are the same in the two algorithms since calculations of u_1 is the same as for t_1 and we will take w_2 and w_3 or $W = w_2 \otimes w_3$ (unfolded to the vector w) as the starting point.

Example of a proof:

At convergence of PLS we have (with W and Q unfolded):

$$t_i = \sum_J x_{iJ} w_J$$
$$q_{L} \propto \sum_i y_{iL} t_i \propto \sum_i \sum_J y_{iL} x_{iJ} w_J$$
$$\sum_{L} q_{L}^2 = 1$$
$$q_{L} = \tau_l \pi_m.$$

Proportionality (α) is used to simplify the expressions. In LTD1 we have

$$t_i = \sum_j \sum_k x_{ijk} w_j w_k$$
$$q_l \propto \sum_i \sum_m y_{ilm} t_i q_m$$
$$\sum_l q_l^2 = 1$$
$$q_m \propto \sum_i \sum_l y_{ilm} t_i q_l$$
$$\sum_m q_m^2 = 1,$$

where there is no need to use q from the previous iteration.

Because $w = w_2 \otimes w_3$ in PLS the steps used to calculate t_1 in PLS and LTD1 are equal. More precisely, let $x_{ijk} = x_{iJ}$ and $w_j w_k = w_J$, then it is immediately seen that the t_i are identical. If it is assumed that $\tau_l = q_l$ it is easy to see that $q_m = \pi_m$ and *vice versa*. It remains only to show that this assumption is true to obtain the complete proof since the calculation of the u and w vectors are isomorphic to the calculation of t and q. But, at convergence the use of q_1 from the previous iteration as a starting guess in step 4a leads to convergence in one iteration of the PCA decomposition of Q. This implies that the assumption is true and the proof is complete.□

Taking into account the property that in PLS t_1 and u_1 have maximum covariance due to the choice of w_J [6] an informative corollary is obtained, namely that 3PLS1c and LTD1 calculates t_1 and u_1 such that they have the maximum co-variance subject to the condition that w_J folded, W, is the product of two vectors, i.e. $W = w_2 \otimes w_3$. The way LTD1 works makes this obvious since the calculation of w_2 is done as a weighted contraction where the weights are w_3 and *vice versa*.