JOURNAL OF CHEMOMETRICS
J. Chemometrics **13,** 491–510 (1999)

CROSS-VALIDATION OF MULTIWAY COMPONENT MODELS

D. J. LOUWERSE,[1] AGE K. SMILDE[1]* AND HENK A. L. KIERS[2]
[1]*Department of Chemical Engineering, Process Analysis & Chemometrics, Nieuwe Achtergracht 166, NL-1018 WV Amsterdam, The Netherlands*
[2]*Department of Psychology, Grote Kruisstraat 2/1, NL-9712 TS Groningen, The Netherlands*

SUMMARY

Two cross-validation methods are presented for multiway component models. They are used for choosing the numbers of components to use in Tucker3 models describing three-way data. The approach is general and can easily be adapted to other three-way and multiway models. A model is estimated after leaving out a small part of the multiway data array. The predictive residual error sum of squares (*PRESS*) is calculated for the eliminated part of the data by comparing the model values with the actual data. *PRESS* of the entire data set can be calculated like this sequentially. The methods are the leave-bar-out cross-validation method, which leaves out data slices in all modes, and the EM cross-validation method, which handles eliminated data as missing values. A method to calculate the statistical significance of the *PRESS* reduction for an additional component, the so called *W*-statistic, is provided for Tucker3 models. A strategy is proposed to search along an efficient path, to reduce computation time, since the number of feasible models as a function of the total number of components summed over the modes increases fast. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: cross-validation; three-way models; Tucker3; PARAFAC; PRESS

1. INTRODUCTION

Many statistical problems occurring in the physical, chemical and chemical engineering sciences can be cast in the form of analyzing data which can be arranged in a matrix (or matrices). A general regression problem, for example, can be formulated as finding a relationship between a univariate y and a (multivariate) $\mathbf{x}$. Estimation of the regression parameters is then performed by a procedure using multiple realizations of y and $\mathbf{x}$ collected in a vector $\mathbf{y}$ and a matrix $\mathbf{X}$. Over the last 10 years another type of data structure has become important in the physical, chemical and chemical engineering sciences: data that can be arranged in a three-way array, or in general when data are arranged in more than three ways, in a multiway array. This calls for specific tools to handle these types of data. Such tools are under development mainly in psychometrics and chemometrics.

A main application area of three-way methods is second-order calibration.[1,2] In separation methods, for instance, second-order data emerge in situations where spectra are collected as a function of the separation time. Another important application area is chemical engineering in the field of multivariate statistical process control (MSPC) of batch processes.[3–5] MSPC can be used for

* Correspondence to: A. K. Smilde, Department of Chemical Engineering, Process Analysis & Chemometrics, Nieuwe Achtergracht 166, NL-1018 WV Amsterdam, The Netherlands.

CCC 0886–9383/99/050491–20 $17.50

Received 20 July 1998
Accepted 22 January 1999

monitoring batch processes, for process optimization and to improve process understanding. Other applications areas are image analysis[6,7] examining complex biochemical systems,[8] multiway calibration in three-dimensional quantitative structure– activity relationships (3D-QSAR) to predict the activity of potential drugs,[9] and modeling a five-factor factorial design of enzymatic activities with a five-way PARAFAC model.[10] Two specialized international meetings on three-way methods have taken place already (in 1993 and 1997), showing that the interest in multiway methods and their application areas is growing.

The tools for handling multiway data are mainly generalizations of principal component analysis (PCA), singular value decomposition (SVD) or factor analysis (FA) of two-way data. Roughly, these generalizations can be divided into parallel factor analysis (PARAFAC) models and Tucker models, where the latter can be subdivided into Tucker1, Tucker2, Tucker3 (three-mode PCA) and, more recently, constrained Tucker models. Kroonenberg[11] explains the Tucker1, Tucker2 and Tucker3 models extensively in an excellent book on this subject. Constrained Tucker models are explained by Kiers, Smilde and co-workers.[12–15] The PARAFAC and Tucker models generalize certain aspects of PCA and FA. The Tucker models are very versatile. Nomikos and MacGregor[3,4] use a Tucker1 model for MSPC, de Ligny *et al*[16] use a Tucker3 model, and when certain constraints are known beforehand, a Tucker2 model or a constrained Tucker3 model can be used. Kiers[12] and Smilde *et al*[13] show the use of PARAFAC and Tucker3 models in second-order calibration, and how to translate specific (chemical) knowledge that is available in constraints on the Tucker3 model. Only specific interactions between the three modes are allowed, simultaneously with non-negativity constraints for specific modes.

One of the major problems of using these methods is determining the number of components to be used. As in PCA[17] the number of components has to be found that optimally separates the systematic part of the data, described by the model, from the residuals, consisting of model errors and measurement errors. Only in specific situations, such as in the case of using constrained Tucker models, is the model size known beforehand.

At present, three-way data are often unfolded into two-way data by taking e.g. frontal slices of a three-way data block and placing them next to each other.[3,4,18] The three-way array of Figure 1 with 'batch', 'variable' and 'time' as modes can be converted to a two-way array with 'batch' and 'variable_time' as modes e.g. by placing two-way data slices, containing all batches and all time periods for every variable, next to each other. The mode variable_time includes all possible combinations of the three-way modes variable and time. The length of this mode is therefore the product of the original three-way modes. Tools suited for handling two-way data, such as PCA and SVD, then can be used. Often these techniques are referred to as unfold PCA (U-PCA), etc. Unfortunately they are sometimes wrongly referred to as multiway PCA, suggesting that the data are analyzed with three-way methods, whereas in principle the data are analyzed with two-way methods.

Obtaining good results for analyzing three-way and multiway data makes it essential to use a model with the appropriate size (as defined by the numbers of components). When this size is not known beforehand, it has to be determined and a proper model has to be found. For this purpose, cross-validation methods are often used in two-way analysis, but not yet in three-way and multiway analysis. Cross-validation uses all the available data to find the optimal model in terms of separating the systematic part of the data, described by the model, from the residuals. Using all available data is especially useful in situations with a limited amount of data. Leaving out objects, for instance to construct a test set, may change the model significantly.

Some of the methods that are currently in use to estimate the size of the three-way models are based on two-way methods. Such methods unfold the three-way data array into three possible matrices. Then for each unfolded matrix the number of components is estimated with PCA or SVD.[18] It can be

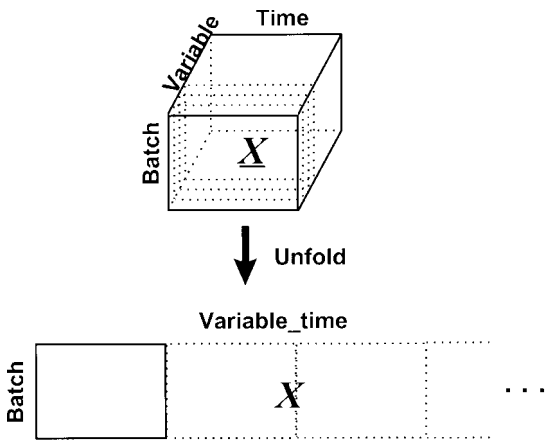

Figure 1. Three-way data $\underline{\mathbf{X}}$ with 'batch', 'variable' and 'time' as modes, unfolded to a two-way matrix $\mathbf{X}$ with 'batch' and combined 'variable_time' as modes

proven straightforwardly that this approach yields the correct number of Tucker3 components when the data are free of noise.[19] An alternative method has been proposed by M. E. Timmerman and H. A. L. Kiers (submitted). This method is based on a systematic comparison of Tucker3 fit values for models with different numbers of components, and searches for a strong decrease in added fit value by additional components. Often, however, (an excessive amount of) noise is present. These approaches can then only be indicative and a true three-way cross-validation method can make an important contribution.

Cross-validation methods for two-way data already exist. Wold[17] and also Eastment and Krzanowsk[20] laid the fundaments for PCA and SVD cross-validation. In principle they leave out parts of the data array, model the remaining data, use the model to estimate the eliminated data, and finally compare the estimated data with the real data. Sequentially all array elements can be eliminated once, and for all elements an estimate can be calculated without using the element itself. In this way the estimates are calculated independently of the data to be estimated, and the predictive power of a model can be determined. The model with the highest predictive power is considered to be the proper model to separate the systematic part of the data from the residuals.

Two alternative generalizations of two-way cross-validation will be presented. One is the approach by Eastment and Krzanowski.[20] This approach will be generalized for multiway data. The other is based on the expectation maximization (EM) approach for handling missing data. In order to do this, all array elements are handled as missing once. The predictive power of the model can be determined by comparing the model value for the missing elements with the real value. Both methods will be worked out for three-way data modeled with Tucker3.

2. THREE-WAY MODELS

Bold capitals refer to two-way arrays or matrices. Bold underlined capitals refer to three-way arrays. Normal characters, including capitals, are scalars and refer to single data elements, or can have a special meaning as will be explained in the text. The 'T' attached to a matrix, as in $\mathbf{X}^{\mathrm{T}}$, refers to the transpose, the '*' in $\mathbf{X}^*$ refers to $\mathbf{X}$ with eliminated data, and the '#' in $\mathbf{X}^{\#}$ refers to the eliminated data themselves.

Figure 2 visualizes the decomposition of the three-way array $\underline{\mathbf{X}}$ with a Tucker3 model. Equation (1)

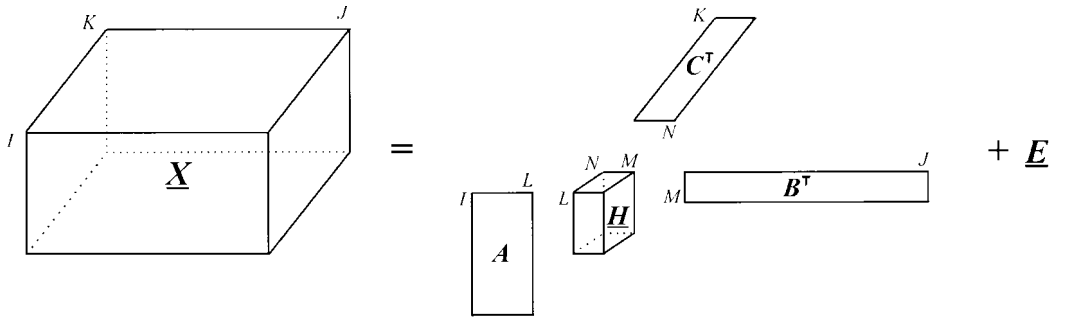

Figure 2. Three-way data $\underline{\mathbf{X}}$ modeled with Tucker3. The modes I, J and K of $\underline{\mathbf{X}}$ respectively are modeled by the matrices **A**, **B** and **C**, with L, M and N components, and a three-way core array $\underline{\mathbf{H}}$ describing all possible component interactions. $\mathbf{B}^{\mathbf{T}}$ and $\mathbf{C}^{\mathbf{T}}$ are the transposes of **B** and **C**, and $\underline{\mathbf{E}}$ is the error array, accounting for model errors and measurement errors

describes the Tucker3 model:

$$x_{ijk} = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} a_{il}\, b_{jm}\, c_{kn}\, h_{lmn} + e_{ijk} \tag{1}$$

where i, j and k are running indices for the units of the three different modes I, J and K respectively, l, m and n are indices for the components for the different modes, and x_{ijk}, a_{ij}, b_{jm}, c_{kn}, h_{lmn} and e_{ijk} are elements of $\underline{\mathbf{X}}$ ($I \times J \times K$), **A** ($I \times L$), **B** ($J \times M$), **C** ($K \times N$), $\underline{\mathbf{H}}$ ($L \times M \times N$) and $\underline{\mathbf{E}}$ ($I \times J \times K$) respectively. The modes I, J and K of $\underline{\mathbf{X}}$ respectively are modeled by the matrices **A**, **B** and **C**, with L, M and N components, and a three-way core array $\underline{\mathbf{H}}$ describing all possible component interactions. $\underline{\mathbf{E}}$ is the error array, accounting for model errors and measurement errors. Often a two-way matrix representation of the Tucker3 model is given by

$$\mathbf{X} = \mathbf{A}\mathbf{H}(\mathbf{C}^{\mathrm{T}} \otimes \mathbf{B}^{\mathrm{T}}) + \mathbf{E} \tag{2}$$

where **X**, **H** and **E** are unfolded two-way representations of $\underline{\mathbf{X}}$, $\underline{\mathbf{H}}$ and $\underline{\mathbf{E}}$ respectively and $\otimes$ is the Kronecker product. The Kronecker product $\mathbf{A} \otimes \mathbf{B}$ ($IK \times JL$) of **A** ($I \times J$) and **B** ($K \times L$) is formed by multiplying each element a_{ij} of **A** by the whole matrix **B**, yielding the supermatrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \dots & a_{1J}\mathbf{B} \\ \vdots & & \vdots \\ a_{I1}\mathbf{B} & \dots & a_{IJ}\mathbf{B} \end{pmatrix}$$

The Tucker3 model can be estimated by fitting these equations with an alternating least squares (ALS) algorithm.[21] PARAFAC, Tucker1, Tucker2 and constrained Tucker models can be described as special cases of Tucker3. For PARAFAC the numbers of components L, M and N are equal in size and only elements of the superdiagonal of $\underline{\mathbf{H}}$ are non-zero. For Tucker2 the third mode is not reduced. This results in a core array $\underline{\mathbf{H}}$ ($L \times M \times K$) and **C** ($K \times K$) that can be joined in a matrix $\tilde{\mathbf{C}}$ ($K \times LM$). Instead of estimating **A**, **B**, **C** and $\underline{\mathbf{H}}$ with ALS, alternatively **A**, **B** and $\tilde{\mathbf{C}}$ can be estimated with ALS. For Tucker1 a PCA is performed on the proper unfolded two-way array of $\underline{\mathbf{X}}$.

3. TWO-WAY CROSS-VALIDATION

The basic principle for estimating the number of components in PCA was first outlined by Wold[17] later improved by Eastment and Krzanowsk[20] and subsequently adjusted by D. J. Louwerse *et al.* (submitted). The concept described by Eastment and Krzanowsk[20] is that for R principal components a predicted value $\hat{x}_{ij}^R$ of x_{ij} is estimated after leaving out a row i ($i = 1,\ldots,I$) and a column j ($j = 1,\ldots,J$) from $\mathbf{X}$ ($I \times J$). The predictive residual error sum of squares (*PRESS*) is calculated by comparing $\hat{x}_{ij}^R$ with x_{ij}:

$$PRESS_R = \sum_{i=1}^{I}\sum_{j=1}^{J}(\hat{x}_{ij}^R - x_{ij})^2 \tag{3}$$

It is essential that $\hat{x}_{ij}^R$ is estimated independently of x_{ij}. That is, the data which are predicted are not used to calculate the model at the same time. The SVD method is used to calculate a PCA model of $\mathbf{X}$. This can be achieved as follows. The SVD of a matrix $\mathbf{X}$ ($I \times J$) is defined by $\mathbf{X} = \mathbf{USV}^{\mathrm{T}}$, where, assuming that $I \geq J$, $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$, $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{VV}^{\mathrm{T}} = \mathbf{I}$, with $\mathbf{I}$ being the identity matrix, and $\mathbf{S}$ is a diagonal matrix with elements $s_{11} \geq s_{22} \geq s_{33} \geq \ldots \geq 0$ being the non-negative square roots of the eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. When R components are considered, the SVD of $\mathbf{X}$ can be written as

$$x_{ij} = \sum_{r=1}^{R} u_{ir}s_{rr}v_{rj} + e_{ij} \tag{4}$$

where x_{ij}, u_{ir}, s_{rr}, v_{rj} and e_{ij} are elements of $\mathbf{X}$ ($I \times J$), $\mathbf{U}$ ($I \times R$), $\mathbf{S}$ ($R \times R$), $\mathbf{V}^{\mathrm{T}}$ ($R \times J$) and $\mathbf{E}$ ($I \times J$) respectively and e_{ij} is the residual of x_{ij} for a model with R components. This model can be used to estimate $\hat{x}_{ij}^R$:

$$\hat{x}_{ij}^R = \sum_{r=1}^{R} u_{ir}s_{rr}v_{rj} \tag{5}$$

When row i is eliminated, the SVD obtained from $\mathbf{X}^{*i}$ ((I-1) $\times J$) gives $\mathbf{U}^{*i}$ ((I-1) $\times R$), $\mathbf{S}^{*i}$ ($R \times R$) and $\mathbf{V}^{*i\mathbf{T}}$ ($R \times J$). Likewise, when column j is eliminated, the SVD obtained from $\mathbf{X}^{*j}$ ($I \times (J$-1)) gives $\mathbf{U}^{*j}$ ($I \times R$), $\mathbf{S}^{*j}$ ($R \times R$) and $\mathbf{V}^{*j\mathbf{Y}}$ ($R \times (J$-1)). $\mathbf{U}^{*j}$ and $\mathbf{V}^{*i}$ are estimates of $\mathbf{U}$ and $\mathbf{V}$ respectively; $\mathbf{S}^{*i}$ $\sqrt{I/(I\text{-}1)}$ and $\mathbf{S}^{*j}$ $\sqrt{J/(J\text{-}1)}$ are both estimates of $\mathbf{S}$. x_{ij} can be estimated by

$$\hat{x}_{ij}^R = \sum_{r=1}^{R} u_{ir}^{*j}\sqrt{s_{rr}^{*j}\sqrt{J/(J-1)}s_{rr}^{*i}\sqrt{I/(I-1)}}v_{rj}^{*j} \tag{6}$$

where $\sqrt{s_{rr}^{*j}\sqrt{J/(J-1)}s_{rr}^{*i}\sqrt{I/(I-1)}}$ is the geometrical mean of both singular values, corrected for the loss of variance in $\mathbf{X}$ by leaving out a row or a column (D. J. Louwerse *et al*, submitted).

In cross-validation it is important to realize that mean centering and/or scaling the data only once, prior to cross-validation, introduces a dependence between x_{ij} and $\hat{x}_{ij}^R$ (D. J. Louwerse *et al*, submitted). If the columns of $\mathbf{X}$ are centered by subtracting the column mean $\bar{x}_j$, and a row is eliminated afterwards, the elements of the eliminated row are used to calculate $\bar{x}_j$, thereby affecting the position of the other rows around the mean. Similarly, scaling columns of $\mathbf{X}$ also will affect the dependence between x_{ij} and x_{ij}^R when a row is eliminated. It is better to center or scale the data after a row or column is eliminated. To make $\hat{x}_{ij}^R$ comparable with x_{ij}, prior to calculating $PRESS_R$, $\mathbf{X}$ also has to be centered and/or scaled. This makes x_{ij} and $\hat{x}_{ij}^R$ completely independent, as it should be.

3.1. *W*-statistic

Often the model with minimum *PRESS* is chosen to be the best model. It is shown by Osten[22] that in many cases the number of components of this model is overestimated. According to Osten[22] the *W*-statistic suggested by Eastment and Krzanowski[20] performs better. It can be used as an *F*-test to determine whether the inclusion of an additional component is significant. Nomikos and MacGregor[3,4] also use this *W*-statistic to determine the number of components in PCA. *W* is defined as

$$W_R = \frac{PRESS_{R-1} - PRESS_R}{D_{\mathrm{F},R}} \div \frac{PRESS_R}{D_{\mathrm{T},R}} \tag{7}$$

where $D_{F,R}$ are the degrees of freedom required to fit the *R*th component and $D_{T,R}$ are the remaining degrees of freedom after fitting *R* components. According to Eastment and Krzanowski[20] W_R represents the quotient of the increase in predictive information supplied by the *R*th component and the average information in each of the remaining components. Note that W_R can become negative; in that case, taking the extra component certainly does not improve the model. The significance of W_R can be tested by comparing the obtained W_R with an $F(D_{F,R}, D_{T,R})$ distribution at e.g. the 5% level. If W_R exceeds $F(D_{F,R}, D_{T,R}; 0{\cdot}95)$, then the *R*th component is considered significant. The degrees of freedom are calculated by considering the number of parameters to be estimated and the constraints on the model at each stage. Caution is needed, for two reasons, when this *W*-statistic is used. First, $PRESS_R$ is not independent of $PRESS_{R-1} - PRESS_R$, as is required for the *F*-test. Secondly, sometimes it is not clear how to calculate the degrees of freedom. For instance, when time series (or spectra) are involved and the sample frequency (or the wavelength grid) is doubled, the number of data points is also doubled. However, according to the Shannon theorem, this does not necessarily mean that the information content is larger. In these cases the true degrees of freedom are not clear.

4. CROSS-VALIDATION OF TUCKER3 MODELS

A three-way data array $\underline{\mathbf{X}}$ of size $I \times J \times K$ can be modeled with a Tucker3 model, resulting in the matrices **A**, **B** and **C** and a three-way core array $\underline{\mathbf{H}}$ (Figure 2). Two cross-validation methods are presented here: first, a cross-validation method that is based on the EM approach to handle missing data; second, a method that leaves out slices of data in every mode.

4.1. Preprocessing three-way data

Before modeling, three-way data are often preprocessed first. This can be necessary to weight data, to get variables on equal footing, to remove offsets, etc. Harshman and Lundy[23] showed that an appropriate way to center a three-way array is to subtract means computed over one-way subarrays (vectors). An appropriate way to rescale three-way data is rescaling *within* one mode, so that for each unit of that mode all elements are multiplied by the same constant. There are a number of possible ways to preprocess three-way data. They are not treated here. However, as in the case of two-way cross-validation, centering and scaling the data only once, prior to cross-validation, can have an effect on the cross-validation result. For this reason the preprocessing method that is used for the original data also has to be used in the cross-validation method, every time a cross-validation step is performed. This ensures the independence between an actual value and a predicted value (see earlier). Moreover, the variability in estimating the means and scaling constants is also incorporated in the cross-validation procedure.

4.2. EM–Tucker3 cross-validation

The EM algorithm is a very general iterative algorithm for estimation of missing data[24] on the basis of model estimates for these data. Since the Tucker3 algorithm is also an iterative algorithm, a combined EM–Tucker3 algorithm can be used for arrays with elements that are (considered) missing. This can be used for cross-validation purposes by considering subsets of randomly chosen elements as missing, and using the actual data values for these elements for validation. The ensuing cross-validation algorithm can be constructed as follows.

1. One or more elements x_{ijk} for $i = 1,\ldots, I$, $j = 1,\ldots, J$ and $k = 1,\ldots, K$ can be chosen at random without replacement from $\underline{\mathbf{X}}$ to form $\underline{\mathbf{X}}^{\#}$. The remaining data $\underline{\mathbf{X}}^{*}$ are used for the model, and the data $\underline{\mathbf{X}}^{\#}$ are handled as missing.
2. $\underline{\mathbf{X}}^{*}$ is preprocessed in a prespecified way
3. Let $\underline{\mathbf{Z}}^{\#}$ be an array of size $\underline{\mathbf{X}}^{\#}$. To determine an estimate of $\underline{\mathbf{X}}^{\#}$, $\underline{\mathbf{Z}}^{\#}$ is filled with sensible starting values, such as the mean values from one of the modes.
4. A Tucker3 algorithm with L, M and N components is used to model the joined array of $\underline{\mathbf{X}}^{*}$ and $\underline{\mathbf{Z}}^{\#}$.
5. After each Tucker3 ALS cycle (Appendix I) the Tucker3 model values $\underline{\hat{\mathbf{X}}}^{\#\,LMN}$ are calculated. $\underline{\mathbf{Z}}^{\#}$ is updated by $\underline{\mathbf{Z}}^{\#} = \underline{\hat{\mathbf{X}}}^{\#LMN}$.
6. Go to step 4 until convergence.
7. Go to step 1 for the next random selection until all elements are eliminated once.
8. Center and/or scale $\underline{\mathbf{X}}$ in the same way as $\underline{\mathbf{X}}^{*}$ (see step 2).
9. Calculate

$$PRESS_{LMN} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(\hat{x}_{ijk}^{LMN} - x_{ijk})^2$$

When the data array is large, selecting more elements simultaneously and handling them as missing speeds up the cross-validation process.

4.3. Leave-bar-out (LBO) Tucker3 cross-validation

In this subsection a model refers to a Tucker3 model with L, M and N components for the three-way loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. Suppose that $I^{\#}$ slices of data are eliminated from mode I of $\underline{\mathbf{X}}$. The remaining data $\underline{\mathbf{X}}^{*i}$ of size $I^{*} \times J \times K$ are preprocessed in the same way as $\underline{\mathbf{X}}$. $\underline{\mathbf{X}}^{*i}$ can be modeled, resulting in $\mathbf{A}^{*}$, $\mathbf{B}_1$, $\mathbf{C}_1$ and $\underline{\mathbf{H}}_1$ (Figure 3). The sizes of $\mathbf{B}_1$, $\mathbf{C}_1$ and $\underline{\mathbf{H}}_1$ are equal to those of $\mathbf{B}$, $\mathbf{C}$ and $\underline{\mathbf{H}}$ and can be considered as estimates of $\mathbf{B}$, $\mathbf{C}$ and $\underline{\mathbf{H}}$. The size of $\mathbf{A}^{*}$ is smaller than that of $\mathbf{A}$. Similarly, $J^{\#}$ slices of mode J are eliminated and $K^{\#}$ slices of mode K. $\mathbf{A}_2$, $\mathbf{C}_2$, $\underline{\mathbf{H}}_2$ and $\mathbf{A}_3$, $\mathbf{B}_3$, $\underline{\mathbf{H}}_3$ are then obtained respectively as estimates of $\mathbf{A}$, $\mathbf{C}$, $\underline{\mathbf{H}}$ and $\mathbf{A}$, $\mathbf{B}$, $\underline{\mathbf{H}}$. The various array sizes are summarized in Table 1. All arrays, except $\mathbf{A}^{*}$, $\mathbf{B}^{*}$ and $\mathbf{C}^{*}$, can be used to estimate the model. In calculating the model with $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{B}_1$, $\mathbf{B}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\underline{\mathbf{H}}_1$, $\underline{\mathbf{H}}_2$, and $\underline{\mathbf{H}}_3$, the data $\underline{\mathbf{X}}^{\#}$ in bar $I^{\#} \times J^{\#} \times K^{\#}$ are not used. Hence, for these eliminated data, predictions $\underline{\hat{\mathbf{X}}}^{\#}{}_{LMN}$ can be obtained using a model whose estimated parameters are independent of these eliminated data. Eight Tucker3 models, listed in Table 2, can be built with the estimates of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ when the estimated cores are assigned to their corresponding component matrices.

An example of such a model according to Tables 1 and 2 is $\underline{\hat{\mathbf{X}}}^{\#}{}_{LMN,5}$, calculated with $\mathbf{A}_3$, $\mathbf{B}_1$ and $\mathbf{C}_1$ and a combined core $\underline{\hat{\mathbf{H}}}$. By analogy with Eastment and Krzanowski,[20] the elements of $\underline{\hat{\mathbf{H}}}$ can be calculated as the geometrical mean of the core matrices $\underline{\mathbf{H}}_3$ (corresponding to $\mathbf{A}_3$) and $\underline{\mathbf{H}}_1$ (corresponding to $\mathbf{B}_1$ and $\mathbf{C}_1$), $\hat{\mathbf{H}} = \sqrt[3]{\mathbf{H}_3 \circ \mathbf{H}_1 \circ \mathbf{H}_1}$, where $\circ$ is the Hadamard product (defined by

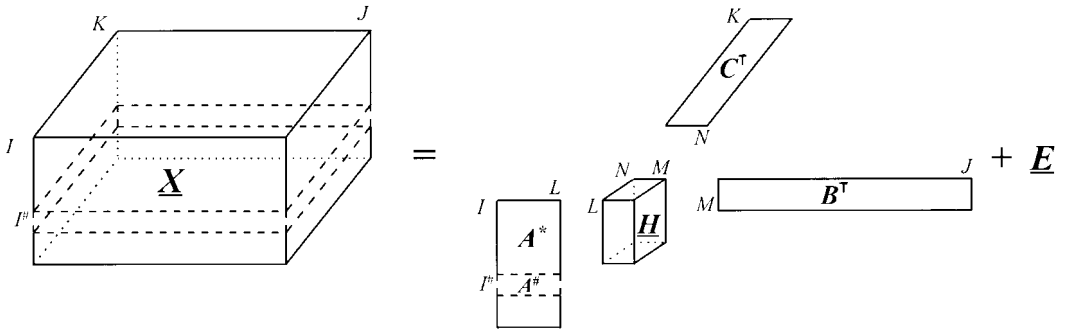

Figure 3. Three-way data modeled with Tucker3. A slice $I^{\#}$ is eliminated from $\underline{\mathbf{X}}$. As a result, the first-mode loading matrix $\mathbf{A}^{*}$ is equally shorter. The modes of $\mathbf{B}$, $\mathbf{C}$ and $\underline{\mathbf{H}}$ are not affected

$[\mathbf{A}\circ\mathbf{B}]_{ij} = a_{ij}b_{ij}$) and the cube root is calculated per element. The average of the eight estimates, $\bar{x}_{ijk}^{LMN}$, is compared with the original data. In this way every element of $\underline{\mathbf{X}}$ can be estimated without using the element itself, and an independent *PRESS* is obtained. To make $\bar{x}_{ijk}^{LMN}$ comparable with x_{ijk}, $\underline{\mathbf{X}}$ has to be centered and/or scaled in the same way as $\underline{\mathbf{X}}^{*}$, prior to calculating $PRESS_{LMN}$ as

$$PRESS_{LMN} = \sum_{i=l}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(\bar{x}_{ijk}^{LMN} - x_{ijk})^2 \tag{8}$$

Three additional problems arise when the estimates are combined. First, the position of a component vector in a loading matrix is not predefined for the Tucker3 algorithm. The first component vector of $\mathbf{A}_2$ is not necessarily comparable with the first component vector of $\mathbf{A}_3$. This problem is dealt with when a Tucker3 algorithm is used with Gram–Schmidt iterations (Appendix I). By this method, principal axes are obtained for $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. The subsequent component vectors in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ describe decreasing amounts of variance. A consequence of using Gram–Schmidt iterations is that $\mathbf{A}^{\mathbf{T}}\mathbf{A} = \mathbf{I}$, $\mathbf{B}^{\mathbf{T}}\mathbf{B} = \mathbf{I}$ and $\mathbf{C}^{\mathbf{T}}\mathbf{C} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The variance of the array is accounted for by $\underline{\mathbf{H}}$, and the loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are always equally normalized. Secondly, the sign of a component is not predefined. The first component vector of $\mathbf{C}_1$ is comparable with plus or minus the first component vector of $\mathbf{C}_2$. This problem can be handled by calculating the inner product of the two comparable component vectors, the sign of which has to be positive. A negative inner product can be corrected by changing the sign of one of the component vectors, while simultaneously compensating for this effect by changing the sign of the corresponding core elements. In this way the model is not changed. Thirdly, when a part of the array is eliminated, the size of the variance is proportionally lower and so is $\underline{\mathbf{H}}$; the variance of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ remains the same since they are

Table 1. Tucker3 models for various array sizes

Array size	Matching Tucker3 model			
$I \times J \times K$	$\mathbf{A}$ $(I \times L)$	$\mathbf{B}$ $(J \times M)$	$\mathbf{C}$ $(K \times N)$	$\underline{\mathbf{H}}$ $(I \times J \times K)$
$I^* \times J \times K$	$\mathbf{A}^*$ $(I^* \times L)$	$\mathbf{B}_1$ $(J \times M)$	$\mathbf{C}_1$ $(K \times N)$	$\underline{\mathbf{H}}_1$ $(I \times J \times K)$
$I \times J^* \times K$	$\mathbf{A}_2$ $(I \times L)$	$\mathbf{B}^*$ $(J^* \times M)$	$\mathbf{C}_2$ $(K \times N)$	$\underline{\mathbf{H}}_2$ $(I \times J \times K)$
$I \times J \times K^*$	$\mathbf{A}_3$ $(I \times L)$	$\mathbf{B}_3$ $(J \times M)$	$\mathbf{C}^*$ $(K^* \times N)$	$\underline{\mathbf{H}}_3$ $(I \times J \times K)$

Table 2. Tucker3 models to estimate $\underline{\mathbf{X}}^{\#}$. The cube root is calculated per element and $\circ$ is the Hadamard product notation

$$\hat{\mathbf{X}}^{\#}_{LMN,1} = \mathbf{A}_2 \sqrt[3]{\mathbf{H}_2 \circ \mathbf{H}_1 \circ \mathbf{H}_1}(\mathbf{B}'_1 \otimes \mathbf{C}'_1)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,2} = \mathbf{A}_2 \sqrt[3]{\mathbf{H}_2 \circ \mathbf{H}_1 \circ \mathbf{H}_2}(\mathbf{B}'_1 \otimes \mathbf{C}'_2)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,3} = \mathbf{A}_2 \sqrt[3]{\mathbf{H}_2 \circ \mathbf{H}_3 \circ \mathbf{H}_1}(\mathbf{B}'_3 \otimes \mathbf{C}'_1)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,4} = \mathbf{A}_2 \sqrt[3]{\mathbf{H}_2 \circ \mathbf{H}_3 \circ \mathbf{H}_2}(\mathbf{B}'_3 \otimes \mathbf{C}'_2)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,5} = \mathbf{A}_3 \sqrt[3]{\mathbf{H}_3 \circ \mathbf{H}_1 \circ \mathbf{H}_1}(\mathbf{B}'_1 \otimes \mathbf{C}'_1)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,6} = \mathbf{A}_3 \sqrt[3]{\mathbf{H}_3 \circ \mathbf{H}_1 \circ \mathbf{H}_2}(\mathbf{B}'_1 \otimes \mathbf{C}'_2)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,7} = \mathbf{A}_3 \sqrt[3]{\mathbf{H}_3 \circ \mathbf{H}_3 \circ \mathbf{H}_1}(\mathbf{B}'_3 \otimes \mathbf{C}'_1)$$

$$\hat{\mathbf{X}}^{\#}_{LMN,8} = \mathbf{A}_3 \sqrt[3]{\mathbf{H}_3 \circ \mathbf{H}_3 \circ \mathbf{H}_2}(\mathbf{B}'_3 \otimes \mathbf{C}'_2)$$

normalized. For this reason the core estimates $\underline{\mathbf{H}}_1$, $\underline{\mathbf{H}}_2$ and $\underline{\mathbf{H}}_3$ are underestimates of $\underline{\mathbf{H}}$. They have to be compensated for this loss of variance, similarly to compensating the singular values $\mathbf{S}$ in equation (6). All core elements are multiplied by the square root of the quotient of the total numbers of slices of the original and the remaining array:

$$\begin{aligned} H_{1,\text{new}} &= H_{1,\text{old}} \sqrt{I/I^*} \\ H_{2,\text{new}} &= H_{2,\text{old}} \sqrt{J/J^*} \\ H_{3,\text{new}} &= H_{3,\text{old}} \sqrt{K/K^*} \end{aligned} \tag{9}$$

5. *W*-STATISTIC FOR TUCKER3

The *W*-statistic for Tucker3 models can be calculated in a similar way as for PCA and SVD (Section 3·2). The *PRESS* values of a model under investigation and a previous model are considered. Suppose an $I \times J \times K$ data array, centered across I, modeled with L, M and N components and a foregoing model with $L-1$, M and N components; W can be calculated as

$$\begin{aligned} W_{LMN,(L-1)MN} &= \frac{PRESS_{(L-1)MN} - PRESS_{LMN}}{D_{LMN,(L-1)MN}} \div \frac{PRESS_{LMN}}{D_{LMN}} \\ D_{LMN,(L-1)MN} &= I + MN - 2L + 1 \\ D_{LMN} &= (I-1)JK - LI - MJ - NK - LMN + L^2 + M^2 + N^2 \end{aligned} \tag{10}$$

The degrees of freedom are calculated as follows[25] Initially there are $I \times J \times K$ degrees of freedom. When data are centered over I, $J \times K$ means are calculated and used to subtract, so $J \times K$ degrees of freedom are lost. There are $LI + MJ + NK + LMN$ parameters used in the model, LI for $\mathbf{A}$, MJ for $\mathbf{B}$, NK for $\mathbf{C}$ and LMN for $\underline{\mathbf{H}}$. However, the lost number of degrees of freedom is smaller than this number. This will be explained for $\mathbf{A}$ ($I \times L$). Since there is rotational freedom, the L vectors of $\mathbf{A}$ can always be rotated such that an identity matrix of size $L \times L$ is formed within $\mathbf{A}$. When $\mathbf{A}$ for example has two vectors,

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} \# & \# & \# & \# & \# \\ \# & \# & \# & \# & \# \end{bmatrix}$$

can be rotated to

$$\tilde{\mathbf{A}}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & \vdots & \tilde{\#} & \tilde{\#} & \tilde{\#} \\ 0 & 1 & \vdots & \tilde{\#} & \tilde{\#} & \tilde{\#} \end{bmatrix}$$

Without changing the model, L^2 parameters can be known *a priori*. The vectors of **A**, **B** and **C** can be rotated simultaneously without changing the model. $L^2 + M^2 + N^2$ parameters of the model are redundant and D_{LMN} is calculated according to equation (10). $D_{LMN,(L\text{-}1)MN}$ is calculated as the difference in degrees of freedom of D_{LMN} and the foregoing model $D_{(L\text{-}1)MN} = (I-1)JK - (L-1)I - MJ - NK - (L-1)MN + (L-1)^2 + M^2 + N^2$. The foregoing model can also have L, $M-1$ and N, or L, M and $N-1$ components. W is then calculated in a similar way. The matching degrees of freedom are

$$\begin{aligned} D_{LMN,L(M-1)N} &= J + LN - 2M + 1 \\ D_{LMN,M(N-1)} &= K + LM - 2N + 1 \end{aligned} \tag{11}$$

An additional component is included in the model when W is significantly larger than one.

6. STRATEGY TO MINIMIZE COMPUTING

Tucker models have numbers of components for each mode. For instance, a (2, 2, 3) model has two components for both the first and the second mode and three components for the third mode. The total number of components in that case is seven. Such a total number of seven components can also be achieved with a (3, 1, 3) model, a (2, 3, 2) model and others. If the model complexity is summarized as the total number of components, then the number of possible combinations rises rapidly with increasing model complexity. Calculating *PRESS* for all possible combinations and subsequently determining the combination of components with lowest *PRESS* is not efficient. A simple strategy which only calculates the *PRESS* values of a limited number of combinations is presented here schematically.

1. *PRESS* is calculated with one component for each mode, equal to a total number of three components.
2. *PRESS* is calculated for three additional models. The former (best) model with L, M, N components is used, and for each mode a model is formed with one additional component. The core sizes of the three new models are $(L+1, M, N)$, $(L, M+1, N)$ and $(L, M, N+1)$. Before calculating *PRESS*, the validity of the model has to be checked as not all combinations of core sizes are sensible. This will be explained in Appendix II[18]
3. Assume the model with lowest *PRESS* to be the best model for the corresponding total number of components.
4. Stop if *PRESS* of the new model is larger than the former model, otherwise go to step 2.

ad 2. Also specific attention has to be paid to branches that are excluded by this procedure. An example of such a branch is (1, 1, 1) → (2, 1, 1) → (2, 1, 2) → (3, 1, 2) → (3, 1, 3) → … Calculating *PRESS* of the models (2, 1, 1) and (3, 1, 2) is redundant (Appendix II). Therefore these branches are not used.

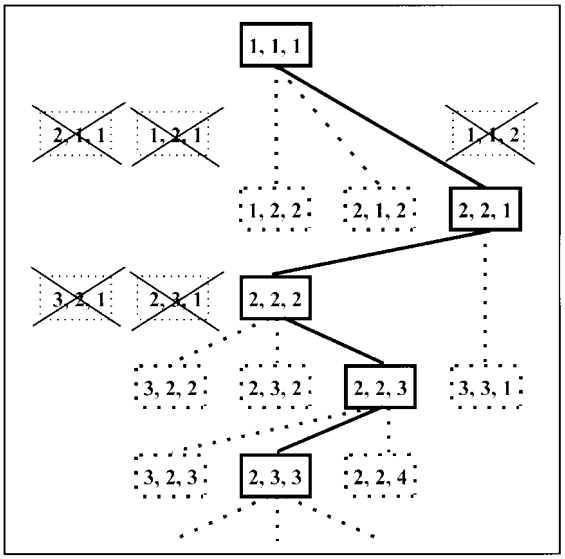

Figure 4. A minimum *PRESS* path used to minimize computing time. The full rectangles denote the minimum *PRESS* track; *PRESS* is also calculated for the broker rectangles. Appendix II explains why *PRESS* is not calculated for the redundant models that are crossed off. The *L*, *M* and *N* Tucker3 component numbers are positioned inside the rectangles

ad 3. Sometimes the *PRESS* values of two models can be close together. It can then be questioned whether the difference is significant. To avoid choosing the wrong 'best' model, both models can be considered as the 'best' model. A tolerance can be used as a criterion.

ad 4. The same argument as for ad 3 is valid for the stop criterion. In order to prevent too fast a termination, again a tolerance can be used.

Figure 4 shows a possible path directed by this procedure. The model with minimum *PRESS* is considered to be the best model. By only selecting the models that are close to the model with minimum *PRESS*, the computing time is considerably reduced. This procedure, however, does not guarantee that the model with the global minimum *PRESS* is found. If there is doubt about the performance of this strategy, the *PRESS* values of additional models have to be calculated.

7. EXAMPLES

Four examples are given to cross-validate Tucker3 models with the presented methods. The data sets used are very different in nature. Both synthetic and real data were used, spanning a range of possible situations encountered in practice. The data sets were as follows.

(a) A noisy synthetic array ($100 \times 20 \times 10$) from M.E. Timmerman and H.A.L. Kiers (submitted). The data are specifically designed with a Tucker3 structure. The core size ($L \times M \times N$) is (3, 3, 3). Elements of the unfolded core (3×9) are [1 0 0, 0 1 0, 0 0 1; 0 1 0, 0 0 1, 1 0 0; 0 0 1, 1 0 0, 0 1 0]. Normally distributed random numbers with zero mean and unit variance are used for **A**, **B** and **C**. The column vectors of **B** and **C** are orthogonalized. 95% of the array variation was described by the model and 5% by normally distributed white noise. A bar size of $10 \times 2 \times 1$ was used for the LBO cross-validation; 20 data points, eliminated at random without replacement, were used for the EM cross-validation. Hence in both cases 0·1% of the data were left out in each cross-validation step. Two-way

cross-validation of the unfolded matrices, based on minimum *PRESS*, properly indicated a (3, 3, 3) core size.

(b) Like (a), with 25% of the array variation described by the model and 75% by noise. Two-way cross-validation of the unfolded matrices, based on minimum *PRESS*, wrongly indicated a (3, 3, 1) core size. This indicates that in cases with a low signal-to-noise ratio the approach of using two-way cross-validation on the three unfolded arrays does not work.

(c) A three-way array from Smilde and Doornbos[26] The logarithms of chromatographic capacity factors (*k*-values) of eight solutes are measured on six stationary phases at six mobile phase compositions. It is the aim to describe differences in stationary phase behavior. The amount of measurement noise is very small in this data set. The three-way array is centered to zero mean over the stationary phase. Both cross-validation methods were performed with leaving out one data point.

(d) A three-way array of size $50 \times 9 \times 200$ from Nomikos and MacGregor[3] The simulated data consist of 50 batches, each with nine variables measured as a function of time (200 time points), describing the normal operating conditions (NOC) of a batch process. It is the aim to monitor deviations from NOC as early as possible in future batches. The array is scaled to zero mean and unit variance over the batches. The data are very noisy; U-PCA can model approximately 30% of the variance in three principal components. A bar size of $5 \times 1 \times 20$ was used for the LBO cross-validation. The EM cross-validation was performed after subdividing the array into units. Each unit consists of 20 subsequent time periods of a specific batch and variable. The $50 \times 9 \times 200$ array was subdivided into $50 \times 9 \times 10$ units. Five units, eliminated at random without replacement, were used for cross-validation. This procedure was applied to avoid problems due to the autocorrelated time series. Information from an eliminated time point k is still largely present in the time points $k+1$ and $k-1$ when a strong autocorrelation exists.

The centering and scaling procedures of the cited examples are performed as described in the literature. Cross-validation was performed in such a way that a new centering and/or scaling procedure was performed when data were eliminated. The synthetic three-way arrays (a) and (b) were used as such; no centering or scaling was performed.

8. RESULTS

PRESS results of relevant models are presented in Table 3 for all examples. Figure 5 shows the minimum *PRESS* path as a function of successive models.

The low-noise synthetic data set (Table 3(a), Figure 5(a)) shows for the EM cross-validation a clear *PRESS* minimum for the (3, 3, 3) Tucker3 model. The *PRESS* value of smaller models is at least twice as large, and the *PRESS* value of larger models either hardly changes or increases slightly. The *W*-statistic confirms that the minimum *PRESS* model is the best model. Thus the correct model is recovered with the EM cross-validation method. The LBO cross-validation also shows a much smaller *PRESS* value for the (3, 3, 3) model compared with smaller models, but not that distinct. For models larger than (3, 3, 3) the *PRESS* value still decreases slightly. According to the *W*-statistic, however, the succeeding models are not significant, as *W* is not significantly larger than one. Especially in this, but also in the other examples presented, it is observed that for comparable models the LBO *PRESS* values are larger than the EM *PRESS* values. This is probably due to the percentage of the data that are used to calculate the individual models. The EM method uses all data points except the eliminated data simultaneously to calculate the model, whereas the LBO method combines several models, individually calculated with eliminated slices. The data that have to be predicted are only a small part of these eliminated slices. Fewer data are used to calculate the individual LBO models compared with the EM model. This probably makes the predictions less precise and is the reason why the LBO *PRESS* values are always larger than the EM *PRESS* values.

Table 3a. Tucker3 model, variance modeled with Tucker3, *PRESS*, *W*-statistic, lost degrees of freedom for latest step ($D_{LMN,\ LMN-1}$) remaining degrees of freedom (D_{LMN}) as a function of Tucker3 models for low-noise synthetic data of size $100 \times 20 \times 10$

Summed components	LBO cross-validation						EM cross-validation					
	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,LMN-1}$	D_{LMN}	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,LMN-1}$	D_{LMN}
0			1252			20000			1252			20000
3	1, 1, 1	74·4	1129	17	128	19872	1, 1, 1	74·4	810·7	85	128	19872
5	2, 1, 2	74·5	813·1	72	107	19765	1, 2, 2	74·5	813·6	[a]	27	19845
6	2, 2, 2	81·1	880·6	[a]	21	19744	2, 2, 2	81·1	622·3	60	101	19744
7	2, 2, 3	81·1	884·8	[a]	9	19735	2, 2, 3	81·1	622·7	[a]	9	19735
8	2, 3, 3	87·6	690·2	265	21	19714	2, 3, 3	87·6	399·8	523	21	19714
9	3, 3, 3	94·6	603·6	27	104	19610	3, 3, 3	94·6	176·1	240	104	19610
10	3, 3, 4	94·6	603·2	1·1	12	19598	3, 3, 4	94·6	176·2	[a]	12	19598
11	4, 3, 4	94·7	600·9	0·7	105	19493	3, 3, 5	94·6	176·2	0	10	19588

[a] The *PRESS* value of the present model is larger than that of the previous model; *W* is negative and not reported.

Table 3b. Tucker3 model, variance modeled with Tucker3, *PRESS*, *W*-statistic, lost degrees of freedom for latest step ($D_{LMN,\ LMN-1}$) and remaining degrees of freedom (D_{LMN} as a function of Tucker3 models for high-noise synthetic data of size $100 \times 20 \times 10$

Summed components	LBO cross-validation						EM cross-validation					
	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,\ LMN\text{-}1}$	D_{LMN}	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,\ LMN\text{-}1}$	D_{LMN}
0			3136			20000			3136			20000
3	1, 1, 1	11·6	2862	15	128	19872	1, 1, 1	11·6	2805	18	128	19872
5	2, 1, 2	12·2	2825	2·4	107	19765	1, 2, 2	11·8	2820	[a]	27	19845
6	2, 2, 2	17·0	2916	[a]	21	19744	2, 2, 2	17·0	2700	8·7	101	19744
7	2, 2, 3	17·1	2926	[a]	9	19735	2, 3, 2	17·1	2698	0·8	19	19725
8	2, 3, 3	21·9	2685	84	21	19714	2, 3, 3	21·9	2529	120	11	19714
9	3, 3, 3	29·7	2565	8·8	104	19610	3, 3, 3	29·7	2295	19	104	19610
10	3, 4, 3	29·9	2562	1·0	22	19588	3, 3, 4	29·8	2299	[a]	12	19598
11	3, 4, 4	30·1	2570	[a]	15	19573	3, 3, 5	29·9	2299	[a]	10	19588

[a] The *PRESS* value of the present model is larger than that of the previous model; *W* is negative and not reported.

Table 3c. Tucker3 model, variance modeled with Tucker3, *PRESS*, *W*-statistic, lost degrees of freedom for latest step ($D_{LMN,LMN-1}$) and remaining degrees of freedom (D_{LMN}) as a function of Tucker3 models for chromatographic data. The order of the Tucker3 model is stationary phase, mobile phase composition and solute ($6 \times 6 \times 8$)

Summed components	LBO cross-validation						EM cross-validation					
	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,LMN\text{-}1}$	D_{LMN}	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,LMN\text{-}1}$	D_{LMN}
0			168·63			240			168·63			240
3	1, 1, 1	98·5	4·18	485	18	222	1, 1, 1	98·5	3·02	676	18	222
5	2, 1, 2	99·1	2·93	8·2	11	211	2, 1, 2	99·1	2·12	8·1	11	211
6	2, 2, 2	99·5	2·15	11	7	204						
7	2, 2, 3	99·6	2·03	1·7	7	197	3, 1, 3	99·1	2·19	[a]	9	202
8	2, 3, 3	99·7	1·97	0·8	7	190	3, 2, 3	99·6	119·00	[a]	12	190
9	2, 4, 3	99·7	1·89	1·6	5	185						

[a] The *PRESS* value of the present model is larger than that of the previous model; *W* is negative and not reported.

Table 3d. Tucker3 model, variance modeled with Tucker3, *PRESS*, *W*-statistic, lost degrees of freedom for latest step ($D_{LMN,LMN-1}$) and remaining degrees of freedom D_{LMN} as a function of Tucker3 models for batch data. The order of the Tucker3 model is batch, variable and time ($50 \times 9 \times 200$)

Summed components	LBO cross-validation						EM cross-validation					
	Model	Var.(%)	*PRESS*	*W*	$D_{LMN,LMN\text{-}1}$	D_{LMN}	Model	Var. (%)	*PRESS*	*W*	$D_{LMN,LMN\text{-}1}$	D_{LMN}
0			88200			88200			88200			88200
3	1, 1, 1	12·8	79359	38	257	87943	1, 1, 1	12·8	77868	45	257	87943
5	2, 1, 2	17·1	77006	11	247	87696	2, 1, 2	17·1	74656	15	247	87696
6	2, 2, 2	19·6	75106	222	10	87686	2, 2, 2	19·6	72854	217	10	87686
7	2, 3, 2	20·5	74899	30	8	87678	3, 2, 2	21·1	72017	21	49	87637
8	3, 2, 3	23·7	73787	5·4	242	87436	3, 2, 3	23·7	69929	13	201	87436
9	4, 2, 3	25·6	72555	30	49	87387	4, 2, 3	25·6	68481	38	49	87387
10	4, 3, 3	26·3	72104	34	16	87371	4, 3, 3	26·3	68003	38	16	87371
11	4, 4, 3	26·5	72132	[a]	14	87357	5, 2, 4	28·8	66748	7·0	234	87137
12	5, 4, 3	27·4	71925	4·7	53	87304	5, 3, 4	29·5	65894	47	24	87113
13	5, 5, 3	27·6	71929	[a]	15	87289	5, 3, 5	31·4	64957	6·1	206	86907
14	5, 6, 3	27·7	71613	30	13	87276	5, 3, 6	32·6	64336	4·1	204	86703
15	5, 7, 3	27·7	71583	3·3	11	87265	6, 3, 6	34·8	62450	46	57	86646
16	5, 8, 3	27·8	71568	2·0	9	87256	6, 4, 6	35·4	62108	13	38	86608
17							7, 3, 7	37·8	60957	7·2	225	86383
18							8, 3, 7	39·1	59388	41	56	86327
19							8, 3, 8	41·1	57517	13	209	86118
20							8, 4, 8	41·7	57421	2·2	66	86052
21							8, 3, 10	43·1	56050	6·1	346	85706

[a] The *PRESS* value of the present model is larger than that of the previous model; *W* is negative and not reported.

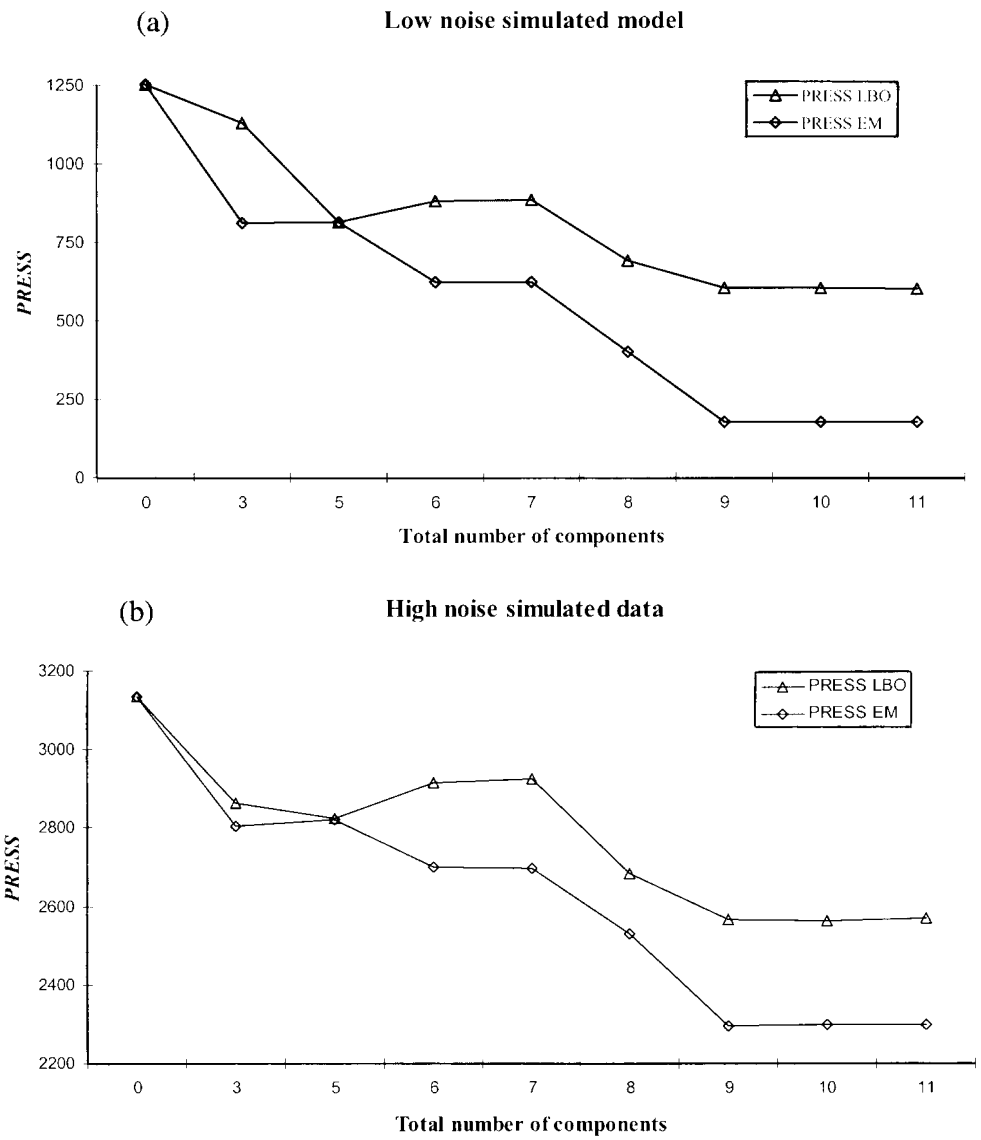

Figure 5. *PRESS* as a function of Tucker3 models for (a) simulated model with a low and (b) with a high noise level, (c) log *k*-values and (d) batch data

The high-noise synthetic data set (Table 3(b), Figure 5(b)) shows the same effect for both methods. With EM cross-validation an absolute minimum at (3, 3, 3) is found. Larger models tend to have a faster increasing *PRESS* value compared with the low-noise data set. The *W*-statistic also confirms that the minimum *PRESS* model is the best model. The LBO cross-validation still shows a slightly smaller *PRESS* value for a larger model (3, 4, 3). However, according to the *W*-statistic, the decrease in *PRESS* value is not significant, so the correct model is found with both methods.

The EM cross-validation method applied to the chromatographic data with capacity factors (Table 3(c), Figure 5(c)) shows a very small *PRESS* value in the case of the (1, 1, 1) model when it is

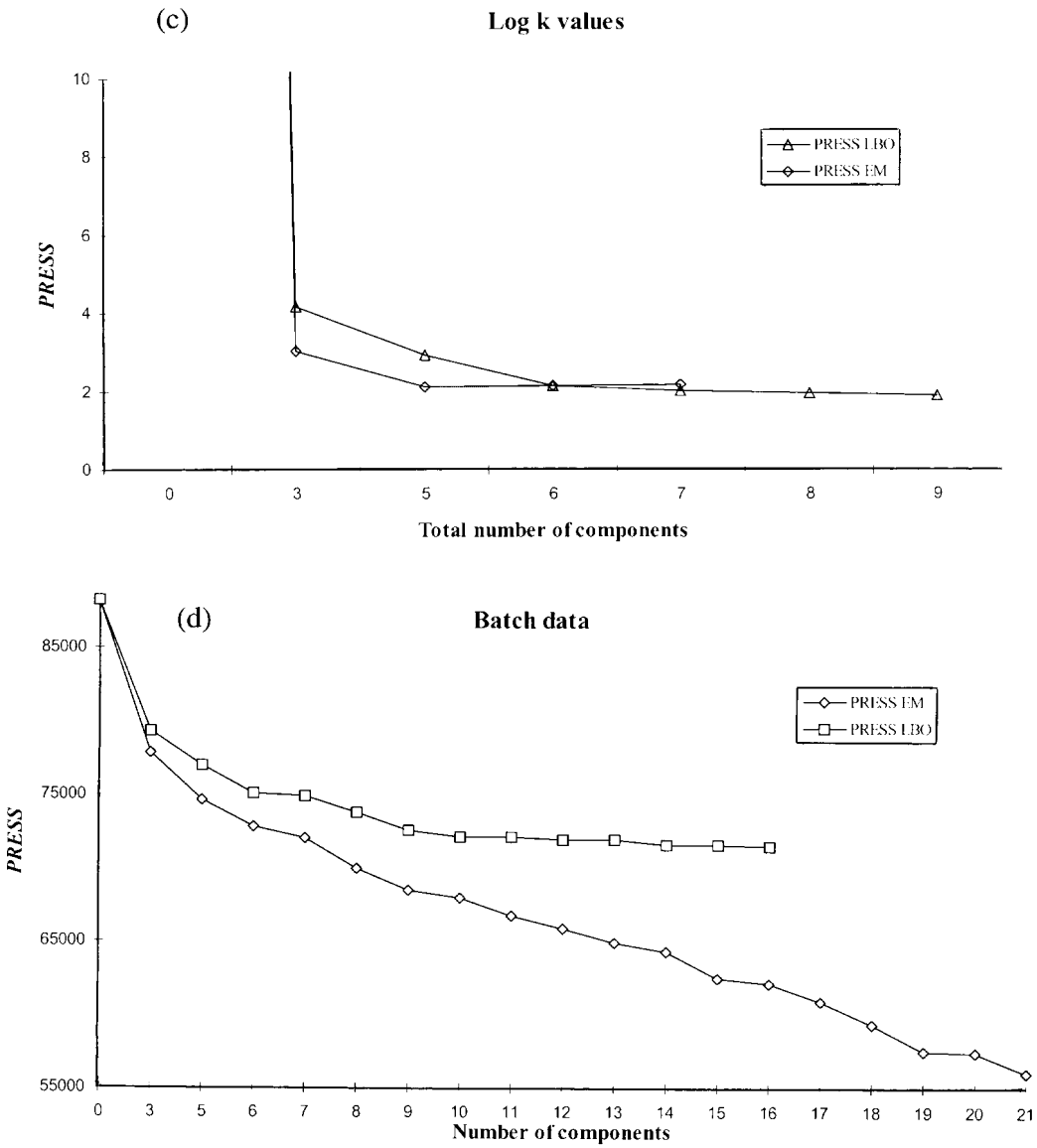

Figure 5. Continued

compared with the total sum of squares. It also shows a minimum *PRESS* value for the (2, 1, 2) model. The *W*-statistic indicates that the (2, 1, 2) model is significant. Therefore the (2, 1, 2) model is the preferred model here. The LBO cross-validation results also show a low *PRESS* value for the smallest possible model. *PRESS* of larger models, however, shows decreasing values. No *PRESS* minimum is found for models up to a total number of nine components. The *W*-statistic of the models (2, 1, 2) and (2, 2, 2) indicates significant models. Larger models are not significant, as *W* is not significantly larger than one. For example, the (2, 2, 3) LBO model shows a *W*-statistic of 1·7, which is below the *F*-test confidence limit of $F(7, 197, 0{\cdot}95) = 2{\cdot}1$. The (2, 2, 2) model is therefore the preferred model for the

LBO cross-validation method. Small models for this data set are also suggested in the literature[26] a one-component unfold PCA model and a one-component PARAFAC model were used. Since there is only one interaction in the (1, 1, 1) Tucker3 model, the one-component PARAFAC model is equal to the (1, 1, 1) Tucker3 model.

Finding the best model for the batch data (Table 3(d), Figure 5(d)) is more troublesome; both methods do not show a minimum. The *PRESS* plot of the LBO method shows a slowly declining plateau starting at the (4, 3, 3) Tucker3 model. *PRESS* of larger models decreases just slightly. The modeled variance also increases slightly for larger models. These results suggest that the (4, 3, 3) Tucker3 model is the preferred model in this case. The EM cross-validation method shows a clear constantly declining *PRESS* plot. The modeled variance shows a constant incline. This suggests that this data set contains a very complicated three-dimensional structure, resulting in a model with a large total number of components, or that a large amount of (correlated) noise is present. On the basis of the *PRESS* plot and the plot of the modeled variance it is difficult to decide which model is the preferred model. A very conservative choice would be the (4, 2, 3) Tucker3 model. Since the time series are correlated, it is difficult to apply the *W*-statistic. Applying the *W*-statistic as if the time series were not correlated did not help in deciding which model to choose. A small reduction in the *PRESS* value nearly always results in a sufficiently large *W*. On the basis of both *PRESS* plots, probably the (4, 2, 3) or the (4, 3, 3) model is preferred.

9. CONCLUSION

In general it can be concluded that cross-validating Tucker3 models with a specific three-way cross-validation method is very well feasible. It is a very useful tool in choosing the proper Tucker3 model size. The two alternative methods, EM cross-validation and LBO cross-validation, perform well for the given examples. The EM cross-validation method can be implemented in software without much difficulty. The LBO cross-validation method is more complex to implement; the requirements, described in Section 4·3, are more difficult to program. In general the EM method outperforms the LBO method; in most cases the *PRESS* plots showed a clearer minimum and it was easier to identify the best model. However, the EM method requires a much longer computation time. The number of Tucker3 models that have to be calculated when a model is validated is much higher. If the size of the eliminated data used in the model is 0·1%, 1000 models have to be calculated with the EM method, whereas 30 models have to be calculated with the LBO method if 10% is eliminated from every mode, which, combined, also pertains to 0·1% ($0{\cdot}1^3$) of the data. The LBO method is much faster and therefore will often be preferable in practice, especially with large data arrays.

There are indications that the 'quick and dirty' method of unfolding the three-way array and using (two-way) PCA on the three unfolded matrices to estimate the number of components in the Tucker models does not work properly for three-way arrays with a high noise level. Both cross-validation methods perform well in this situation.

Considering the time-intensive calculations, a strategy for finding the best model speeds up the process considerably. As mentioned before, this strategy does not guarantee that the best model is found. Specific models that might be important can be excluded by the procedure. Therefore this procedure is useful only as a tool in helping to find the best model.

The *W*-statistic can be of help in choosing the best model. Especially for (small) arrays with independent data without autocorrelation, the *W*-statistic can be very informative in determining the best cross-validated model. For (large) arrays with time series or other autocorrelated data, a visual inspection of the *PRESS* plot as a function of the model size will have to do. In this case the number of subsequent points in the autocorrelated mode has to be chosen such that it is large compared with the

time constant of the autocorrelation function. As the degrees of freedom are not known in this case, it is difficult to apply the *W*-statistic.

The principles of the presented EM and LBO cross-validation approaches can easily be generalized for other three-way and multiway models. PARAFAC models, for instance, can be validated similarly to Tucker3 models, as they can be regarded as restricted Tucker models with elements present only on the superdiagonal of the core.

APPENDIX I: TUCKER3 ALGORITHM WITH GRAM–SCHMIDT ITERATIONS

The meanings of the variables are in accordance with Figure 2 and equation (1). Starting values of the component matrix **A** are obtained by first unfolding $\underline{\mathbf{X}}$ to a matrix of size $I \times JK$, then performing an SVD on this matrix and defining **A** to be the first L singular vectors. Starting values of **B** and **C** are similarly obtained by unfolding $\underline{\mathbf{X}}$ to $J \times IK$ and $K \times IJ$ respectively. For $\underline{\mathbf{H}}$, its optimal least squares value, given the current values for **A**, **B** and **C**, when fitting the Tucker3 model to the data is used

Then alternating least squares is performed by updating **A** given **B**, **C** and $\underline{\mathbf{H}}$, updating **B** given **A**, **C** and $\underline{\mathbf{H}}$, updating **C** given **A**, **B** and $\underline{\mathbf{H}}$, and updating $\underline{\mathbf{H}}$ given **A**, **B** and **C**, until convergence, by means of the Gram–Schmidt-based updating procedure proposed in Reference 27. After convergence, **A** contains eigenvectors of $\mathbf{X}(\mathbf{C}\mathbf{C}^{\mathbf{T}} \otimes \mathbf{B}\mathbf{B}^{\mathbf{T}})\mathbf{X}^{\mathbf{T}}$, put in order of decreasing eigenvalue. **B** and **C** contain eigenvectors of analogous matrices.

APPENDIX II: INVALID TUCKER3 MODELS

In Tucker3 not all core sizes are sensible. The rank of the various matrices restricts the core size to $L \leq MN$, $M \leq LN$ and $N \leq LM$.[19] Specific combinations violating these inequalities can give the same results as other combinations with a smaller core (a lower total number of components), hence these models are redundant. An example of such a redundant model is a model with $L = 2$, $M = 1$, $N = 3$ (2, 1, 3). The maximum rank of **C**, equal to N, is two ($N \leq LM = 2$). Therefore this model and the (2, 1, 2) model will model the data equally well.

REFERENCES

1. E. Sanchez and B. R. Kowalski, *J. Chemometrics*, **2**, 265 (1988).
2. A. K. Smilde, P. H. van der Graaf, D. A. Doornbos, T. Steerneman and A. Sleurink, *Anal. Chim. Acta*, **235**, 41 (1990).
3. P. Nomikos and J. F. MacGregor, *AIChEJ.* **40**, 1361 (1994).
4. P. Nomikos and J. F. MacGregor, *Technometrics*, **37**, 41 (1995).
5. K. A. Kosanovich, K. S. Dahl and M. J. Piovoso, *Ind. Engng. Chem. Res.* **35**, 138 (1996).
6. P. Geladi, H. Isaksson, L. Lindqvist, S. Wold and K. Esbensen, *Chemometrics Intell. Lab. Syst.* **5**, 209 (1988).
7. P. Geladi and K. Esbensen, *J. Chemometrics*, **5**, 97 (1991).
8. S. Leurgans and R. T. Ross, *Statist. Sci.* **7**, 289 (1992).
9. J. Nilsson, S. de Jong and A. K. Smilde, *J. Chemometrics*, **11**, 511 (1997).
10. R. Bro and H. Heimdal, *Chemometrics Intell. Lab. Syst.* **34**, 85 (1996).
11. P. M. Kroonenberg, *Three-mode Principal Component Analysis: Theory and Applications*, DSWO Press, Leiden (1983).
12. H. A. L. Kiers, *Statist. Appl.* **4**, 659 (1992).
13. A. K. Smilde, Y. Wang and B. R. Kowalski, *J. Chemometrics*, **8**, 21 (1994).
14. A. K. Smilde, R. Tauler, J. M. Henshaw, L. W. Burgess and B. R. Kowalski, *Anal. Chem.* **66**, 3345 (1994).
15. H. A. L. Kiers and A. K. Smilde, *J. Chemometrics*, **12**, 125 (1998).
16. C. L. de Ligny, C. Spanjer, J. C. van Houwelingen and H. M. Weesie, *J. Chromatogr.* **301**, 311 (1984).
17. S. Wold, *Technometrics*, **20**, 397 (1978).
18. S. Wold, P. Geladi, K. Esbensen and J. Öhman, *J. Chemometrics*, **1**, 41 (1987).
19. L. R. Tucker, *Psychometrika*, **31**, 279 (1966).
20. H. T. Eastment and W. J. Krzanowski, *Technometrics*, **24**, 73 (1982).

21. P. M. Kroonenberg and J. De Leeuw, *Psychometrika*, **45**, 69 (1980).
22. D. W. Osten, *J. Chemometrics*, **2**, 39 (1988).
23. R. A. Harshman and M. E. Lundy, *Research Methods for Multimode Data Analysis. Data Preprocessing and the Extended PARAFAC Model*, p. 216, Praeger, New York (1984).
24. R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, Chap. 7, Wiley, New York (1987).
25. H. M. Weesie and J. C. van Houwelingen, *GEPCAM User's Manual*, Institute of Mathematical Statistics, Utrecht (1983).
26. A. K. Smilde and D. A. Doornbos, *J. Chemometrics*, **5**, 345 (1991).
27. P. M. Kroonenberg, J. M. F. Ten Berge, P. Brouwer and H. A. L. Kiers, *Comput. Statis. Q.* **5**, 81 (1989).

