ELSEVIER

Chemometrics and Intelligent Laboratory Systems 42 (1998) 105–113

Chemometrics and intelligent laboratory systems

Improving the speed of multiway algorithms
Part II: Compression

Rasmus Bro [*], Claus A. Andersson [1]

Chemometrics Group, Food Technology, Department of Dairy and Food Science, Royal Veterinary and Agricultural University, Rolighedsvej 30, iii, DK-1958 Frederiksberg C, Denmark

Abstract

In this paper an approach is developed for compressing a multiway array prior to estimating a multilinear model with the purpose of speeding up the estimation. A method is developed which seems very well-suited for a rich variety of models with optional constraints on the factors. It is based on three key aspects: (1) a fast implementation of a Tucker3 algorithm, which serves as the compression method, (2) the optimality theorem of the CANDELINC model, which ensures that the compressed array preserves the original variation maximally, and (3) a set of guidelines for how to incorporate optional constraints. The compression approach is tested on two large data sets and shown to speed up the estimation of the model up to 40 times. The developed algorithms can be downloaded from http:\\newton.mli.kvl.dk\foodtech.html. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Tucker3; PARAFAC; CANDELINC; Constraints; Tucker1; Data compression

1. Introduction

An annoying aspect of estimating some multiway models using alternating least squares (ALS) is the time consumption of these algorithms. A way to increase the speed of ALS algorithms is to compress the data array initially and then subsequently estimate the model from the compressed data. This is natural since a multiway model is per se a compression of the original data into fewer parameters, implying that the systematic variation in the data is expressible in less than the original number of data points. Hence, the model to be estimated should also be estimable from another (condensed) representation of the systematic variation in the data. Furthermore, since a multiway model can be considered a multilinear decomposition preserving the systematic variation in the data, it seems useful to use a multilinear decomposition for compression as well. After estimating the parameters of the model in the compressed space, these can then be transformed to the original space, and hopefully provide a good approximate solution to the solution that would be found if estimating the model directly from the raw data. In the sequel we will refer to the model used to compress the data as the *compression model* and the model operating on the compressed array as the *analytical model*.

Alsberg and Kvalheim have described in a series of papers a method for compressing high dimen-

[*] Corresponding author. E-mail: rb@kvl.dk.

[1] E-mail: ca@kvl.dk.

PII: S0169-7439(98)00011-2

sional arrays [1,2]. Kiers and Harshman [3] have shown that this approach is equivalent to the CANDELINC (CANonical DEcomposition with LINear Constraints) approach. In CANDELINC, only orthonormal bases are allowed but any non-orthonormal basis can be orthogonalized prior to compression without any loss of information [4]. The Alsberg and Kvalheim approach was developed specifically for estimating Tucker3 models, while the CANDELINC approach is valid for estimating any multiway model. Furthermore as stressed by Kiers and Harshman [3] there is no need for special algorithms in the CANDELINC approach. One simply regresses the data onto the bases, use any existing multiway algorithm on the compressed array, and decompress the result by premultiplying the solution with the projection bases. This, however, only holds for unconstrained models with a nonweighted least squares optimization criterion as will be shown. The only important constraint that does not require any special attention is orthogonality. If orthogonal loadings are found in, e.g., a PARAFAC model of the compressed array, then the backtransformed solution will also be orthogonal. In this paper Tucker3 is suggested for finding the compression bases as the Tucker3 algorithm is very fast and has the property of providing optimal bases in a least squares sense. Alsberg and Kvalheim suggest different bases in their work. *If* the size of the array is so large that estimation of the Tucker3 model is in practice impossible due to the computer capacity, then these suggested bases are sensible, but if the computer capacity is sufficient it is not sensible to use other bases than those defined by the Tucker3 model.

Note that the suggested compression approach is relevant for estimating most multiway models. Even in the case where one is merely interested in a Tucker3 model, compressing the array first, enables one to quickly estimate models of different dimensions and perhaps using different constraints in order to find the most appropriate model.

The compression method is developed and evaluated on several data sets. It is shown that the new method makes multiway modeling faster and more memory-efficient. It is discussed how to express important constraints and weighting schemes in the modeling of compressed arrays. Three-way arrays will be used as an example but the developed theory is directly applicable for arrays of any order. The ALS procedure for estimating the PARAFAC model will be used throughout but the method is also applicable for other models and algorithms.

In the following, scalars are indicated by (lowercase) italics, vectors by bold lower-case characters, bold capitals are used for two-way matrices, and underlined bold capitals for three-way arrays. The ijkth element of $\underline{\mathbf{X}}$ is called x_{ijk} and is the element in the ith row, jth column, and kth tube of $\underline{\mathbf{X}}$. When three-way arrays are unfolded to matrices, the following notation will be used: if $\underline{\mathbf{X}}$ is an $I \times J \times K$ array and is unfolded to an $I \times JK$ matrix, $\mathbf{X}$, the order of J and K indicates which indices of J are running fastest. In this case the indices of J are running fastest, meaning that the first J columns of $\mathbf{X}$ contain all variables for $k=1$ and for $j=1$ to $j=J$. For short we will define the $I \times JK$ matrix $\mathbf{X}^{(1)}$ where the superscript indicates that it is the *first* mode that is preserved. Likewise $\mathbf{X}^{(2)}$ is a $J \times IK$ matrix and $\mathbf{X}^{(3)}$ a $K \times IJ$ matrix. If the arrangement of the array is clear from the context the superscript will not be shown.

2. Theory

An $I \times J \times K$ array $\underline{\mathbf{X}}$ is given. Suppose that the rank of the systematic variation in each of the three modes is R^{A}, R^{B}, and R^{C}, respectively. By the rank of the systematic variation is meant the minimum rank of an appropriate basis for the space spanned by the systematic variation in a particular mode, i.e., the rank if no noise was present [5]. For the first mode the rank of, and a basis for, the variable space can for example be determined from analyzing the $I \times JK$ unfolded matrix $\mathbf{X}$ obtained by concatenating the K layers of size $I \times J$ of $\underline{\mathbf{X}}$ one after another.

Several methods exist for determining the proper rank, e.g., judging the residuals, using cross-validation [6,7] or methods similar to Malinowski's indicator function [8]. For compression, however, it is not essential to find the exact rank, but rather to define the rank so large that the correct rank is less than the defined rank. Let $\mathbf{U}$ of size $I \times R^{\mathrm{A}}$ be an orthonormal basis for the space spanned by systematic variation in the first mode. An orthonormal matrix $\mathbf{V}$ of size $J \times R^{\mathrm{B}}$ similarly defines the variable space of the systematic variation in the second mode and an orthonormal matrix $\mathbf{Z}$ of size $K \times R^{\mathrm{C}}$ defines the variable space in the third mode. An F-component PARAFAC model is sought for the $I \times J \times K$ array $\underline{X}$. An F-component PARAFAC model is defined through $\mathbf{A}$ $(I \times F)$, $\mathbf{B}$ $(J \times F)$, and $\mathbf{C}$ $(K \times F)$ as

$$\min \left\| \sum_{k=1}^{K} \mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}} \right\|_F^2 \tag{1}$$

where $\mathbf{X}_k$ is the kth layer of $\underline{X}$, i.e., the $I \times J$ matrix obtained by fixing the third mode at its kth value. The matrix $\mathbf{D}_k$ is a diagonal matrix containing the kth row of $\mathbf{C}$ in its diagonal. General information on the PARAFAC model can be found in many papers [9–13]. As the optimal $\mathbf{A}$ is approximately describing the systematic variation in the first mode of $\underline{X}$ it must hold that a matrix exists such that

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Gamma}, \tag{2}$$

as $\mathbf{U}$ is a basis for the systematic variation. Similar relations hold for the second and third mode:

$$\mathbf{B} = \mathbf{V}\boldsymbol{\Theta} \tag{3}$$

and

$$\mathbf{C} = \mathbf{Z}\boldsymbol{\Xi}. \tag{4}$$

This is the same as saying, that the PARAFAC model is linearly constrained to the subspaces $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$. The CANDELINC model was developed for estimating multiway models under such linear constraints [4]. The theory of the CANDELINC model states that if a PARAFAC model of $\underline{X}$ given by $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ is sought, subject to the above constraints, then it is only necessary to estimate the (much) smaller matrices $\boldsymbol{\Gamma}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Xi}$. More importantly these matrices can be found by estimating a PARAFAC model on an array $\underline{Y}$ of size $R^{\mathrm{A}} \times R^{\mathrm{B}} \times R^{\mathrm{C}}$ found by regressing $\underline{X}$ onto the orthonormal bases $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$. Written in matrix notation letting $\mathbf{X}$ be the $I \times JK$ unfolded array, and $\otimes$ denoting the Kronecker product these regressions read

$$\mathbf{Y}^{(1)} = \mathbf{U}^{\mathrm{T}}\mathbf{X}^{(1)}(\mathbf{Z} \otimes \mathbf{V}). \tag{5}$$

Estimating an F-component PARAFAC model of $\underline{Y}$ will give the loading matrices $\boldsymbol{\Gamma}$ $(R^{\mathrm{A}} \times F)$, $\boldsymbol{\Theta}$ $(R^{\mathrm{B}} \times F)$, and $\boldsymbol{\Xi}$ $(R^{\mathrm{C}} \times F)$, and through the relations of Eqs. (2)–(4) the loading matrices in the original spaces can be calculated.

If the orthonormal bases are bases for the systematic variation, then the model estimated from $\underline{Y}$ (Eq. (5)) will give the sought solution. In Ref. [4] this is shown for any model that can be regarded as a Tucker3 model or a restricted version of a Tucker3 model. The PARAFAC, PARATUCK2 [14], PARAFAC2 [15,16], and the Tucker2 [17] models can all be regarded as restricted versions of Tucker3 and can hence be estimated from the compressed array without loss of information under the constraints of Eqs. (2)–(4).

The crucial point in this method is to find good bases for the respective modes. If these are appropriate, one would expect the analytical model estimated from the compressed space to be equal to the model estimated from the raw data. One possibility for finding these bases would be to use the singular vectors from a singular value decomposition (Tucker1) of the array properly unfolded for each direction. That is, $\mathbf{U}$ would equal the first R^{A} left singular vectors from an SVD of $\mathbf{X}^{(1)}$. The bases $\mathbf{V}$ and $\mathbf{Z}$ are found similarly. From these estimated bases and the relation in Eq. (5) the compressed array can be determined. In short for the Tucker1-based compression one obtains the projections matrices as

Tucker1-based compression

$$\begin{aligned} [\mathbf{U},\mathbf{S},\mathbf{T}] &= \mathrm{svd}(\mathbf{X}^{(1)}, R^{\mathrm{A}}) \\ [\mathbf{Z},\mathbf{S},\mathbf{T}] &= \mathrm{svd}(\mathbf{X}^{(2)}, R^{\mathrm{B}}) \\ [\mathbf{V},\mathbf{S},\mathbf{T}] &= \mathrm{svd}(\mathbf{X}^{(3)}, R^{\mathrm{C}}), \end{aligned} \tag{6}$$

where the function $[\mathbf{R},\mathbf{S},\mathbf{T}] = \mathrm{svd}(\mathbf{X},F)$ calculates the rank F truncated singular value decomposition of the matrix $\mathbf{X}$, the matrix $\mathbf{R}$ holding the first F left singular vectors. Note that this approach has actually been suggested earlier for the PARAFAC model specifically in Ref. [18].

A better way, though, to define optimal bases is to say that $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$ should give a least squares estimate of the array $\underline{Y}$ of Eq. (5). This will lead to a set of bases which preserves most of the original variation in the compressed array. The definition of the array $\underline{Y}$ in Eq. (5) corresponds to the definition of the so-called core array in a Tucker3 model [19]. It therefore immediately follows that orthonormal loading matrices of a $R^{\mathrm{A}} \times R^{\mathrm{B}} \times R^{\mathrm{C}}$ Tucker3 model will provide optimal bases for calculating the compressed array. Further, the compressed array will be equal to the core array of the Tucker3 model. Realizing this, it then follows that a fast Tucker3 model is the key to a successful compression method. In part I [20], such an algorithm was developed for the MATLAB programming environment. After obtaining the array $\underline{Y}$ any suitable model can be estimated as described in Refs. [3,4], and exemplified above for the PARAFAC model.

Tucker3-based compression

$$\underset{\mathbf{U},\mathbf{V},\mathbf{Z},\mathbf{Y}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{U}\mathbf{Y}(\mathbf{Z}^{\mathrm{T}} \otimes \mathbf{V}^{\mathrm{T}})\|_F^2 \tag{7}$$

It is important that most systematic variation be incorporated into the compressed array. This is especially true if the subsequent analytical model to be estimated is constrained in some sense. Henceforth, the goal of the Tucker3 model is not to find *the* model but rather to find a model that is not underestimated with respect to dimensions. It is of little concern whether the compressed array is of size 7 × 7 × 7 or 11 × 11 × 11 with respect to the speed of the algorithm, but it may have a significant influence on the quality of the model if not all systematic variation is retained in the 7 × 7 × 7 array. In general, very few data types conform exactly to a mathematical model, which means that one must expect some systematic variation in the residuals. If, e.g., a three-component PARAFAC model is sought it will not necessarily be sufficient to compress the array using a 3 × 3 × 3 Tucker3 model. The way to choose the appropriate number of components in the compression model depends highly on the data type. No general rules can be given, unless one is willing to settle for a quite large compressed array. One may for example compress the array using, say five extra components compared to the number of components in the analytical model, which would probably ensure a valid model. If this is not satisfactory, one has to resort to numerical rank-analysis or simply evaluate for increasing number of components in the compression model when the estimated final model no longer changes. The results presented in Ref. [21] as well as here indicate that using the same number of factors in the Tucker3 model as in the subsequent analytical model will work satisfactory in many cases though not all.

2.1. Modifications of the compression approach

In the literature algorithms have been given for estimating the three-way PARAFAC and Tucker3 model in situations where only one mode is very high-dimensional [22,23]. These methods are exact and implicitly based on the fact, that the rank of the high-dimensional mode is limited by the dimensionalities of the remaining modes. If the product, *d*, of the two smallest dimensions of the array is smaller than the dimension in the mode of the largest size, then it can be shown that the rank of this mode is upper-bounded by *d*. In the present approach this means that in situations with one very high-dimensional mode, one can simply compress only in the high-dimensional mode using a basis of dimension *d*. This will provide a compressed array that exactly preserves the variation of the original array (A.K. Smilde, personal communication). It can be shown that such a compression model can be estimated by a Tucker1 model.

In general, if some modes are not to be compressed this is implemented in the compression method by estimating a Tucker2 or a Tucker1 model instead of the Tucker3 model. Avoiding compression in a certain mode can be useful, e.g., if the mode is to be estimated with constraints that do not easily translate into the compressed space.

If the uncertainties (e.g., standard deviations) of the individual elements are known, several possibilities exist for incorporating these uncertainties in the loss function of the analytical model. One may scale the data prior to compression [24–26] or compute the compression model using a weighted alternating least squares regression approach. The analytical model can henceforth be estimated with an unweighted loss function. Instead of using the uncertainties in the compression model, one may also simply estimate the compression model without considering these. The uncertainty of the elements of the *compressed* array may then be obtained by regressing the uncertainties (same size as $\underline{X}$) in the same way as $\underline{X}$ is regressed (Eq. (5)). These uncertainties can hence be used when estimating the analytical model.

If the data array contains missing values, the compression must be performed taking this into account as described in Part I [20]. The resulting compressed array will have no missing entries and hence no special attention is needed in the algorithm for estimating the analytical model.

If the resulting loading matrices of the analytical model are required to be nonnegative this poses some problems, as the bounded least squares problem of the uncompressed problem turns into a more general and complicated inequality constrained least squares problem in the compressed space. Currently no method seems able to handle this special situation efficiently but the problem is being worked on, and will be the subject of a following paper.

3. Experimental

Two data sets arising from fluorescence spectroscopy were used for testing the compression on real data. The first called AMINO is a data set of five samples with different amounts of tryptophane, phenylalanine, and tyrosine. Each sample has been measured spectrofluorometrically at excitation 250–300 nm, emission 250–450 nm with 1 nm intervals. The data have also been described in Ref. [13]. The data array is of dimension 5 (sample) × 51 (excitation) × 201 (emission). Note that for these data the exact compression mentioned before is not possible. Even though the first mode is only of dimension 5, the product of the two smaller is 255 which is more than the largest dimension. The proper PARAFAC-dimensionality of the data has been found to be three. The other data set stems from an investigation of a sugar plant process and is called SUGAR. It suffices here to say that 265 samples of sugar were dissolved in water and measured spectrofluorometrically from 275–560 nm at excitations 230, 240, 255, 290, 305, 325, 340 nm by a procedure according to Nørgaard [27]. Part of the data was significantly influenced by Rayleigh scatter. In order not to confound the results with the problems of missing values, this part of the data set was discarded in this analysis resulting in an array of size 265 × 371 × 7. The proper PARAFAC dimensionality is three.

For the AMINO data set, the following procedure was used. The unconstrained PARAFAC model was estimated for a two-, three-, and four-component model respectively. This way it is possible to judge separately what happens if the model is under- or over-specified with respect to the number of components. For the SUGAR data, only a three-component PARAFAC model was estimated. A relative change in fit (sum-of-squared errors) less than 10^{-6} was used as convergence criterion. Each model was estimated from the raw data and from an array compressed using two and up to seven components in the Tucker3 model to verify the influence of the degree of compression. Naturally one expects that the fewer components in the Tucker3 model, the faster the subsequent estimation will be as the array is smaller. However, one will also expect that the estimated analytical model resembles the model estimated from the

Table 1
Results from estimating a two-component PARAFAC model on the data set AMINO

Data set AMINO, two-component PARAFAC model					SVD-based component	
Tucker3 components	Time of comp/ model (s)	Time of model (s)	Time of raw (s)	Difference (‰ experimental)	Time of comp/ model (s)	Difference (‰ experimental)
2	97	43	324	$-0.6 \cdot 10^{-3}$	147	8.4
3	16	8	324	$6.3 \cdot 10^{-3}$	144	$6.8 \cdot 10^{-3}$
4	37	11	324	$2.4 \cdot 10^{-3}$	147	$2.4 \cdot 10^{-3}$
5	65	16	324	$1.9 \cdot 10^{-3}$	149	$2.0 \cdot 10^{-3}$
6	146	17	324	$1.6 \cdot 10^{-3}$	152	$1.9 \cdot 10^{-3}$
7	73	19	324	$0.4 \cdot 10^{-3}$	155	$0.4 \cdot 10^{-3}$

The first column gives the number of components used in the Tucker3 compression. The second column is the time spent in estimating both the compressed array and the model. The third column gives the time for only estimating the model from the compressed array, and the fourth column the time for estimating the model from the raw data. The fifth column gives the difference in the percentage of variation explained by the two models. The last two columns give the results from compressing with Tucker1 loadings.

Table 2
Results from estimating a three-component PARAFAC model on the data set AMINO

Data set AMINO, three-component PARAFAC model					SVD-based compression	
Tucker3 components	Time of comp/ model (s)	Time of model (s)	Time of raw (s)	Difference (‰ experimental)	Time of comp/ model (s)	Difference (‰ experimental)
3	32	24	849	$2.0 \cdot 10^{-5}$	151	$2.3 \cdot 10^{-4}$
4	56	29	849	$7.7 \cdot 10^{-5}$	157	$1.1 \cdot 10^{-4}$
5	82	35	849	$4.1 \cdot 10^{-5}$	162	$0.4 \cdot 10^{-4}$
6	165	41	849	$3.4 \cdot 10^{-5}$	168	$0.4 \cdot 10^{-4}$
7	101	48	849	$1.3 \cdot 10^{-5}$	175	$0.1 \cdot 10^{-4}$

For further explanation, see legend of Table 1.

raw data better, the more components are used for compression.

The time used for estimating the model is given in seconds and three times are tabulated: the time used for estimating the model from the raw data (using the same initialization as for the Tucker3 model), the time used for compression and estimating the model from the compressed array and finally the time used for estimating the analytical model from the compressed array. The last one is relevant as one will often estimate different models from the data in order to verify which is better. In such a case one would not recompress the array each time, but rather use the same compressed array each time. We have chosen to use time rather than the number of FLOPS (floating operations) for indicating the computational complexity, as the number of FLOPS seldom reflects the time consumption realistically. In order to be able to generalize the results obtained to other platforms than MATLAB, however, we will also mention the complexity of the methods with respect to FLOPS. The difference in fit between the model estimated from the raw data and from the compressed data is also given. The model estimated from the raw data is the ‘truth’ as it will per definition be the least-squares estimate; hence the fit of the model estimated from the compressed data, should give equally good fit.

For comparison, the results of using Tucker1-based compression instead of a Tucker3 is also shown. These Tucker1-defined bases are often suggested as appropriate bases for describing the respective variable spaces in the literature. Indeed, if differ-

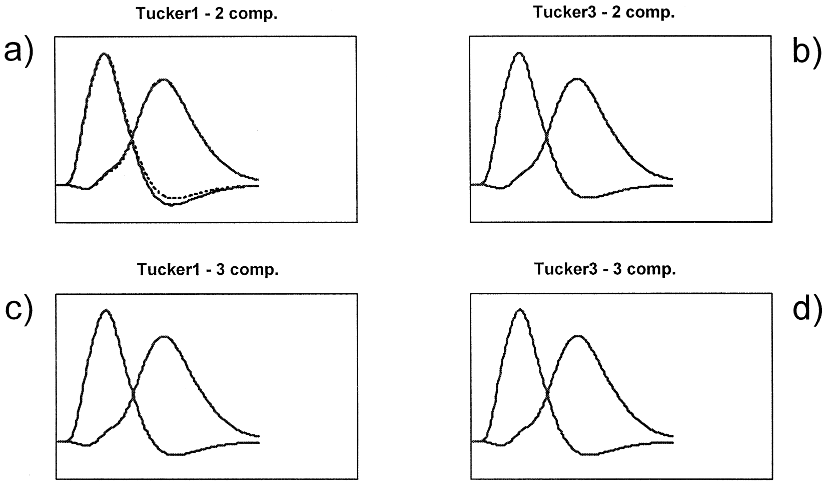

Fig. 1. Two-component PARAFAC model of AMINO. The broken lines indicate the loadings estimated directly from the raw data. (a) Using two-component Tucker1 for compression, (b) using two-component Tucker3 for compression, (c) using three-component Tucker1 for compression, (d) using three-component Tucker3 for compression.

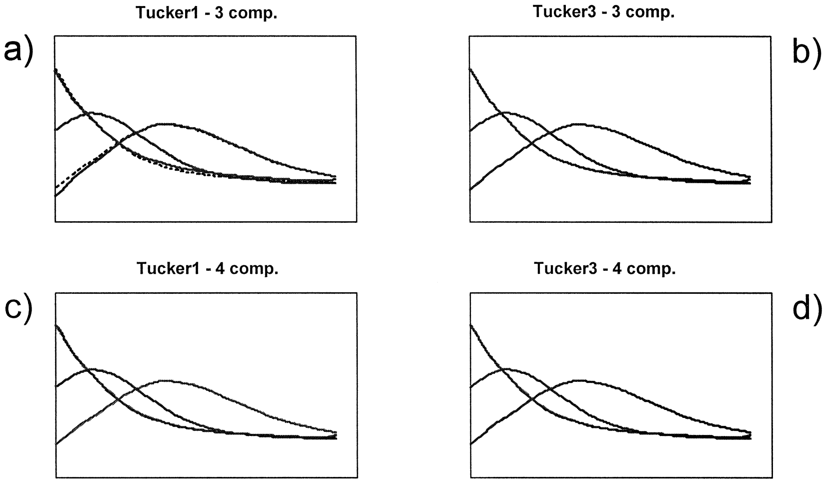

Fig. 2. Three-component PARAFAC model of SUGAR. The broken lines indicate the loadings estimated directly from the raw data. (a) Using three-component Tucker1 for compression, (b) using three-component Tucker3 for compression, (c) using four-component Tucker1 for compression, (d) using four-component Tucker3 for compression.

ences in time and fit between these two compression approaches are negligible there is little sense in using the more complicated iterative Tucker3 model for compression.

4. Results

The most important finding of the investigation is that the analytical model obtained from the compressed data is almost always identical to the one obtained from the raw data. Of all the models estimated only two compression based analytical models differ substantially from the models estimated directly from the raw data. These are the Tucker1-based models shown in Tables 1 and 2 with two- and three-compression components, respectively. To illustrate qualitatively the difference between the Tucker1- and the Tucker3-based compression the estimated loadings in the emission mode are compared in Figs. 1 and 2 with the loadings estimated from the raw data.

The estimates are shown for the models mentioned above, and models including one more component in the compression bases. It is easily verified that only Tucker1-based models differ from the reference loadings (Fig. 1aFig. 2a). Using more components will remedy this (Fig. 1cFig. 2c) and the Tucker3-based compression is always better (Fig. 1b and dFig. 2b and d). The overall conclusion as judged from the tables is, that Tucker3 compressed PARAFAC modeling is consistently faster than uncompressed modeling. Especially if the PARAFAC

Table 3
Results from estimating a four-component PARAFAC model on the data set AMINO

Data set AMINO, four-component PARAFAC model					SVD-based compression	
Tucker3 components	Time of comp/ model (s)	Time of model (s)	Time of raw (s)	Difference (‰ experimental)	Time of comp/ model (s)	Difference (‰ experimental)
4	874	580	1130	$14.0 \cdot 10^{-3}$	989	$14.0 \cdot 10^{-3}$
5	627	581	1130	$4 \cdot 10^{-3}$	669	$4 \cdot 10^{-3}$
6	671	547	1130	$2 \cdot 10^{-3}$	760	$4 \cdot 10^{-3}$
7	625	573	1130	$0 \cdot 10^{-3}$	710	$1 \cdot 10^{-3}$

For further explanation, see legend of Table 1.

Table 4
Results from estimating a three-component PARAFAC model on the data set SUGAR

Data set SUGAR, three-component PARAFAC model					SVD-based compression	
Tucker3 components	Time of comp/model (s)	Time of model (s)	Time of raw (s)	Difference (‰ experimental)	Time of comp/model (s)	Difference (‰ experimental)
3	307	111	1545	$5.4 \cdot 10^{-3}$	172	$64.8 \cdot 10^{-3}$
4	297	123	1545	$5.4 \cdot 10^{-3}$	172	$8.4 \cdot 10^{-3}$
5	447	150	1545	$0.5 \cdot 10^{-3}$	174	$0.5 \cdot 10^{-3}$
6	455	176	1545	$0.4 \cdot 10^{-3}$	177	$0.4 \cdot 10^{-3}$
7	450	202	1545	$0.2 \cdot 10^{-3}$	180	$0.2 \cdot 10^{-3}$

For further explanation, see legend of Table 1.

model is slightly overparameterized (too many components) the gain is large, as the estimation of the PARAFAC model from the raw data can then be very time-consuming (Table 3). Surprisingly, modeling based on Tucker3 compression is also faster than using the simpler Tucker1-based compression. This is because the Tucker1 estimation of bases is performed on quite large arrays. This could have been remedied by using instead an approach similar to the initialization of the Tucker3 algorithm as described in part 1. The Tucker3 compression though, consistently fits the reference model better than Tucker1-based compression. Especially if few compression components are used the difference can be large (Tables 1 and 4). There are thus no arguments for using Tucker1-based compression instead of Tucker3-based.

For all the models investigated the number FLOPS used for estimating the models were also registered. The main result is that estimating the PARAFAC model using the Tucker3-based compression is generally 5 to 80 times cheaper than estimating the model from the raw data in terms of FLOPS as compared to only a 3 to 40 times cheaper with respect to speed. Much of the computation (30–90%) is used for estimating the Tucker3 model, though with respect to time these figures are generally lower. Even though the compression approach is thus advantageous any improvement of the Tucker3 algorithm will be beneficial. A simple idea could be to only estimate the Tucker3 compression model using few iterations ($<$ 10). The observation that the Tucker1 based approach is almost as efficient as the Tucker3 based approach seems to indicate, that even an approximate Tucker3 model can be beneficial. However, one must keep in mind, that for all practical purposes, several analytical models will normally be estimated, but only one compression model is needed. Therefore the actual importance of the complexity of the compression algorithm is less important than indicated by the results presented here.

5. Conclusion

We have developed an efficient method for compressing large arrays using a fast Tucker3 algorithm for compression. The compression method has been shown to speed up estimation considerably. Incorporation of important constraints has also been discussed. It might be argued that there is little gain in using Tucker3 loadings instead of the more easily calculated Tucker1 models for compression. However, as the Tucker3 model *is* fast and because it *does* make a difference in some situations, the use of Tucker3 loadings seems an appropriate choice for compression. This, especially since the estimation of the compression model is mostly fast compared to the estimation of the possibly several analytical models. The conclusion of this work also applies to, e.g., the use of singular vectors for defining the variable space before doing generalized rank annihilation.

Acknowledgements

Professor Lars Munck is thanked for providing philosophical support for what must be the most exploratory industry-related research group. The authors further wish to thank Professor Lars Munck for financial support through the Nordic Industry Foundation project P93149 and the FØTEK foundation. We are obliged to Age Smilde, Sijmen de Jong, Henk Kiers, and the anonymous referees for helpful comments.

References

[1] B.K. Alsberg, O.M. Kvalheim, Chemometrics Intell. Lab. Syst. 24 (1994) 31.
[2] B.K. Alsberg, O.M. Kvalheim, Chemometrics Intell. Lab. Syst. 24 (1994) 43.
[3] H.A.L. Kiers, R.A. Harshman, Chemometrics Intell. Lab. Syst. 39 (1997) 31.
[4] J.D. Carroll, S. Pruzansky, J.B. Kruskal, Psychometrika 45 (1980) 3.
[5] V. Tomišić, V. Simeon, J. Chemometrics 7 (1993) 381.
[6] S. Wold, Technometrics 20 (20) (1978) 397.
[7] H. Martens, T. Næs, Multivariate Calibration, Wiley, Chichester, 1989.
[8] E.R. Malinowski, Factor Analysis in Chemistry, Wiley, New York, 1991.
[9] R.A. Harshman, UCLA Work. Pap. Phonetics 16 (1970) 1.
[10] R.A. Harshman, UCLA Work. Pap. Phonetics 22 (1972) 111.
[11] S. Leurgans, R.T. Ross, R.B. Abel, SIAM J. Matrix Anal. Appl. 14 (1993) 1064.
[12] R.A. Harshman, M.E. Lundy, Comp. Stat. Data Anal. 18 (1994) 39.
[13] R. Bro, Chemometrics Intell. Lab. Syst. 38 (1997) 149.
[14] R.A. Harshman, M.E. Lundy, Psychometrika 61 (1996) 133.
[15] R.A. Harshman, UCLA Work. Pap. Phonetics 22 (1972) 30.
[16] H.A.L. Kiers, Psychometrika 54 (1989) 515.
[17] P.M. Kroonenberg, Three-Mode Principal Component Analysis, DSWO Press, Leiden, 1989.
[18] C.J. Appellof, E.R. Davidson, Anal. Chem. 53 (1981) 2053.
[19] P.M. Kroonenberg, J. de Leeuw, Psychometrika 45 (1980) 69.
[20] C.A. Andersson, R. Bro, Chemometrics Intell. Lab. Syst. 42 (1998) 91–101.
[21] H.A.L. Kiers, J. Chemometrics (submitted).
[22] H.A.L. Kiers, W.P. Krijnen, Psychometrika 56 (1991) 147.
[23] H.A.L. Kiers, Psychometrika 57 (1992) 415.
[24] J.M.F. ten Berge, in: R. Coppi, S. Bolasco (Eds.), Multiway Data Analysis, Elsevier, North-Holland, 1989.
[25] J.B. Kruskal, Proc. Symp. Appl. Math. 28 (1983) 75.
[26] R.A. Harshman, M.E. Lundy, in: H.G. Law, C.W. Snyder, J.A. Hattie, R.P. McDonald (Eds.), Research Methods for Multimode Data Analysis, Praeger, New York, 1984.
[27] L. Nørgaard, Zuckerindustrie 120 (1995) 970.