JOURNAL OF CHEMOMETRICS
J. Chemometrics 2002; **16**: 198–205
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cem.709

‘Stretch’ vs ‘slice’ methods for representing three-way structure via matrix notation

Richard A. Harshman* and Sungjin Hong
Department of Psychology, University of Western Ontario, London, Ontario N6A 5C2, Canada

Received 2 January 2001; Accepted 14 September 2001

A three-way array must be represented in two-way form if its structure is to be described and manipulated by means of matrix notation. Historically, two methods, here called ‘array stretching’ and ‘array slicing’, have been used. More recently, however, array slicing has often been overlooked, resulting in a loss of mathematical flexibility. ‘Stretching’ involves matricizing (unfolding) the three-way array and applying one’s mathematical operations to the resulting two-way matrix; this results in expressions that are often quite useful for parameter estimation but which are relatively long and require practice to interpret properly. ‘Slicing’ involves taking a representative two-way subarray and applying operations to it; this often gives compact and easily understood expressions but requires the introduction of extra matrix names and becomes awkward if the array is not ‘slicewise regular’. In this paper the advantages of each approach are demonstrated and compared by applying them to a set of models from the Tucker and Parafac families. In addition, we show how slicewise representation can be improved by using (i) angle brackets to eliminate the need for extra diagonal matrices, and (ii) ‘encapsulated summation’ notation to allow representation of array structure that is orderly but not slicewise regular. Copyright © 2002 John Wiley & Sons, Ltd.

KEYWORDS: diagonalization; sliced array; stretched array; matricized; unfolding; multilinear and quasi-multilinear models; Tucker; Parafac/Candecomp; Parafac1; Parafac2; Paratuck2

1. INTRODUCTION

Tucker developed the first three-way factor analysis in the 1960s [1–3]. In order to work with three-way arrays, he had to overcome the basic restriction of matrix notation to two-way arrays. To do so, he developed several methods of representing three-way structure and expressing the application of linear transformations to that structure. One method was to use a scalar expression (usually a sum of products of model parameters) to stand for a ‘representative’ array element. This approach produces statements of models that are often relatively easy to understand and hence it continues to be used by many researchers when introducing a multilinear model.

The main problem with the ‘representative element’ approach is that the scalar expression cannot easily be used in a matrix equation or product. Hence it is not well suited for expressing the application of linear transformations such as rotations of co-ordinate systems in the latent factor spaces.

However, Tucker developed two matrix-based methods of representing three-way structure that are fully compatible with standard matrix multiplication.[‡] The first method was the use of a matrix expression (involving one or more subscripted matrices) to stand for the structure of a representative array slice [1,2,4]. By implication, the structure it described applied to all slices and hence it described the entire array. For simplicity we will call this the ‘array slice’ or simply the ‘slice’ approach. Tucker’s second method was to use the full three-way array, but only after converting it into two-way form [3,4] by what has been called ‘unfolding’ or ‘matricizing’ [5]. He referred to the two-way converted array as a two-mode matrix with one ‘elementary mode’ and one ‘combination mode’ [3]. As a contrasting metaphor to ‘slice’, we will call this method array ‘stretching’ in this paper, but we are not proposing it as a general alternative or replacement to ‘unfolding’ or ‘matricizing’.

Thus Tucker provided the foundations for both ‘slice’ and ‘stretch’ representations of three-way structure. More recently, however, the ‘slice’ notation has been somewhat overlooked. We believe this results in a loss of mathematical flexibility and simplicity. The purpose of this paper is to review the above-mentioned developments and, through the use of examples, to compare the different methods of representing three-way structure. The examples will show that sometimes the ‘stretch’ notation is best but other times the ‘slice’ notation is to be preferred. In addition, we propose a couple of improvements in the ‘slice’ notation to make it more transparent and effective.

The examples which we use to assess and/or demonstrate the different methods of representing three-way structure

*Correspondence to: R. A. Harshman, Department of Psychology, University of Western Ontario, London, Ontario N6A 5C2, Canada.
E-mail: harshman@uwo.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada; Contract/grant number: OGP-000-7896.

[‡] He also defined a special product ‘$\leftarrow\!\!\ominus$’ to allow matrix array multiplication ‘in the depth mode’ [2] and a matrix array product ‘$\times_n$’ [3], but these will not be considered here.

Table I. Scalar representation of eight (quasi) multilinear models

	Model name	Representative array element
1	Tucker3	$x_{ijk} = \sum_r \sum_s \sum_t a_{ir} b_{js} c_{kt} g_{rst}$
2	Tucker2	$x_{ijk} = \sum_r \sum_s a_{ir} b_{js} g_{rsk}$
3	Two versions of Tucker's TMMDS/Carroll's IDIOSCAL	$y_{ii'k} = \sum_r \sum_{r'} a_{ir} h_{rr'k} a_{i'r'}$
		$y_{ii'k} = \sum_r \sum_{r'} a_{ir} c_{kr} \phi_{rr'k} c_{kr'} a_{i'r'}$
4	Tucker1	$x_{ijk} = \sum_r a_{ir} g_{rjk}$
5	Parafac1	$x_{ijk} = \sum_r a_{ir} b_{jr} c_{kr}$
6	Indirect fit Parafac1(or orthogonal Parafac2)	$y_{ii'k} = \sum_r a_{ir} c_{kr}^2 a_{i'r}$
7	Parafac2	$y_{ii'k} = \sum_r \sum_{r'} a_{ir} c_{kr} \phi_{rr'} c_{kr'} a_{i'r'}$
8	Paratuck2	$x_{ijk} = \sum_r \sum_s a_{ir} \, {}^{\mathrm{A}}c_{kr} \, h_{rs} \, {}^{\mathrm{B}}c_{ks} \, b_{js}$

(1) The familiar T3 model. (2) The T2 model, which only determines factors for two of the three data modes. (3) Known as either Tucker's three-mode multidimensional scaling (TMMDS) or Carroll's IDIOSCAL model, this model fits a structure for a set of symmetric matrices such as covariance matrices or scalar-product matrices; both authors also propose the second more detailed and informative version in which the varying factor interactions given by the h_k are decomposed into factor weights and (cosines of) angles between factors. (4) The T1 model, which determines factors for only one of the three modes. (5) The familiar Parafac/Candecomp model [12,13]. (6) The version of Parafac/Candecomp which extracts orthogonal factors by applying Parafac1 to symmetric matrices such as covariances, or, equivalently, the version of Parafac2 which is constrained to orthogonal factors. (7) The Parafac2 model for fitting covariances subject to the constraint that inter-factor cosines are fixed. (8) The Paratuck2 model which includes PARAfac-like uniqueness [9] and TUCKer-like interactions among factors.

are eight multilinear or quasi-multilinear models of recent interest in the chemometrics literature (e.g. in Reference [6]). The names and the 'representative element' expressions of each model are given in Table I. Three (rows 1, 2 and 4) are from the Tucker family [1–3,6,7] and three (rows 5–7) are from the Parafac family [6,8], while the models in rows 3 and 8 bridge both [9–11] (also J. D. Carroll and J.- J. Chang, paper presented at the meeting of the Psychometric Society, Princeton, NJ, March 1972).

2. THE TWO METHODS

2.1. Array slicing

In his first formulation of the three-way model [1], Tucker used matrix predefinition and 'stand-in' matrices to state the model in matrix form. Starting with the representative element formulation (model 1, Table I), he rephrased the scalar expression as

$$x_{ijk} = \sum_r \sum_s a_{ir} b_{js} \sum_t c_{kt} g_{rst} \tag{1}$$

then defined

$$n_{rsk} = \sum_t c_{kt} g_{rst} \tag{2}$$

to produce the matrix formulation of T3 (Reference [1], p. 127) as

$$\mathbf{X}_k = \mathbf{A}\mathbf{N}_k\mathbf{B}' \tag{3}$$

Not just the statement of the model, but also rotation, array preprocessing, etc. were discussed using array slice representation, including cases in which the subscripts were indices of latent, rather than surface, distinctions. The main problem with this notation was that it was not completely transparent, because it required the predefinition of $\mathbf{N}_k$, which obscured the presence of $\mathbf{C}$ and $\underline{\mathbf{G}}$ elements.

2.2. Array stretching

Tucker subsequently discovered a way to construct a single matrix expression for T3 built simply from the four parameter sets **A, B, C** and $\underline{\mathbf{G}}$ [3]: he 'matricized' (unfolded) the three-way array so that it became a partitioned two-way matrix, then represented the structure of this matrix by using a combination of standard matrix multiplication and a Kronecker product of the parameter matrices. This would initially lead to the following expression for the T3 model:

$$[\mathbf{X}_1|\mathbf{X}_2|\cdots|\mathbf{X}_K] = \mathbf{A}[\mathbf{G}_1|\mathbf{G}_2|\cdots|\mathbf{G}_T](\mathbf{C}'\otimes\mathbf{B}') \tag{4}$$

where '$\otimes$' represents the Kronecker product. This is not so compact as (3), but the whole array is represented and no predefined matrix is required. Note that whether Equation 4 should be written using $\mathbf{C}'\otimes\mathbf{B}'$ or $\mathbf{B}'\otimes\mathbf{C}'$ depends on whether one uses the 'right' or the 'left' direct or Kronecker product (Reference [14], p. 81; cf. Reference [15], p. 55). Use of the right product is more common, but not universal (see e.g. Reference [16], p. 145). For simplicity, given all the other novel notation, we did not use the left Kronecker product in this paper; nonetheless, we suggest that it be considered in the future. For one thing, it would restore agreement with the usual convention that the fastest-changing index in a set is the leftmost one. This would avoid conflicts of convention such as in (5) below, where on the left side of the equal sign we place the fastest-changing index, *j*, before *k*, but on the right we must reverse this order because of the Kronecker product convention that the fastest-changing index is on the right-hand matrix.

3. NOTATION IMPROVEMENTS FOR MATRICIZED ARRAYS

3.1. Tucker's notation for unfolded arrays

As Equation 4 demonstrates, a disadvantage of array stretching is that it can produce quite long expressions. Tucker proposed a notational convention to make it more compact, by attaching special pre- and post-subscripts to the matrix names (Reference [3], p. 282). Using his notation, a simple $I \times R$ matrix $\mathbf{A}$ is represented as ${}_i\mathbf{A}_r$, for example. Reversing the index symbols in the pre- and post-subscripts gives the transpose, so that if $\mathbf{B} = {}_j\mathbf{B}_r$ then $\mathbf{B}'$ is ${}_r\mathbf{B}_j$. A product such as $\mathbf{AB}'$ is represented by suppressing the repeated inner matrix dimension and writing it as ${}_i\mathbf{A}_r\mathbf{B}_k$. Finally, a matricized array is written with multiple index letters in either the pre- or post-subscript position to indicate the combination mode. For example, ${}_i\mathbf{X}_{(jk)}$ represents the partitioned array on the left side of (4). Thus, Equation 4 can be rewritten more compactly as

$${}_i\mathbf{X}_{(jk)} = {}_i\mathbf{A}_r\mathbf{G}_{(st)}({}_t\mathbf{C}_k \otimes {}_s\mathbf{B}_j) \tag{5}$$

3.2. Bro's superscript unfolding specification

A different notational convention has been recently suggested by Bro [6]. Also quite compact, it conforms more closely to standard matrix notation than Tucker's. This version provides information about the unfolding in a parenthesized (and unitalicized) uppercase superscript to the array name. For example, the horizontally unfolded array ${}_i\mathbf{X}_{(jk)}$ in (5) becomes $\mathbf{X}^{(\mathrm{I \times JK})}$. Likewise, for $\underline{\mathbf{X}}$ unfolded vertically we have

$$\mathbf{X}^{(\mathrm{IK \times J})} = \begin{bmatrix} \mathbf{X}_1 \\ \hline \mathbf{X}_2 \\ \hline \vdots \\ \hline \mathbf{X}_K \end{bmatrix} \tag{6}$$

and for horizontally adjoined transposes we have

$$\mathbf{X}^{(\mathrm{J \times IK})} = [\mathbf{X}_1'|\mathbf{X}_2'|\cdots|\mathbf{X}_K'] \tag{7}$$

Bro's superscript-based unfolding notation also generalizes in a natural way to situations not considered by Tucker. For example, a four-way array might be unfolded or matricized as

$$\mathbf{X}^{(\mathrm{J \times IKL})} = [\mathbf{X}_{1,1}'|\mathbf{X}_{2,1}'|\cdots|\mathbf{X}_{K,L}'] \tag{8}$$

or as

$$\mathbf{X}^{(\mathrm{IK \times JL})} = \left[\begin{array}{c|c|c} \mathbf{X}_{1,1} & \mathbf{X}_{1,2} & \cdots \\ \hline \mathbf{X}_{2,1} & \mathbf{X}_{2,2} & \cdots \\ \hline \vdots & \vdots & \ddots \end{array}\right] \tag{9}$$

In (9), both modes are combination modes; this might be useful for certain theoretical purposes. It is easy to see how unfoldings such as (6)–(9) can be represented, although less concisely, by extending Tucker's format. However, this is not the case for higher-way array structures such as $\underline{\mathbf{X}}^{(\mathrm{IK \times JL \times NM}\ldots)}$.

Using this 'superscript unfolding notation', the T3 model can be written as

$$\mathbf{X}^{(\mathrm{I \times JK})} = \mathbf{AG}^{(\mathrm{R \times ST})}(\mathbf{C}' \otimes \mathbf{B}') \tag{10}$$

which deviates less from standard matrix conventions than (5).

3.3 Kiers' simplification

Taking things one step further, Kiers [5] has recently proposed a shorthand that can sometimes replace Bro's notation with something even simpler. To use it, the unfolding of the matrix must follow certain mode-order conventions. Briefly, the conventions are that (i) there is always one elementary mode and it determines the rows, and (ii) in the combination mode the original array modes are ordered sequentially, except for modes whose labels alphabetically precede that of the elementary mode; these occur sequentially at the end. (For a more precise definition in terms of mode-order permutation, and helpful examples, see Reference [5].) For unfoldings that follow these conventions, the shorthand notation provides an elegant, concise specification.

However, these conventions impose some non-trivial restrictions. For example, even though the unfolding patterns in (6)–(9) seem natural, Kiers' notation cannot represent them and so the superscript notation must be used. Equation 10 is suitable, however, and it is by far the most common case.

The Kiers shorthand indicates a matricized array by a small non-italicized subscript containing the name of the elementary mode, e.g.,

$$\mathbf{X}_{\mathrm{a}} = [\mathbf{X}_1|\mathbf{X}_2|\cdots|\mathbf{X}_K] \tag{11}$$

This is sufficient to define the mode structure of the matrix provided that it has been matricized to fulfil the order convention. Likewise,

$$\mathbf{X}_{\mathrm{c}} = \begin{bmatrix} (\mathrm{vec}(\mathbf{X}_1))' \\ \hline (\mathrm{vec}(\mathbf{X}_2))' \\ \hline \vdots \\ \hline (\mathrm{vec}(\mathbf{X}_K))' \end{bmatrix}$$

However, greater flexibility and uniformity are often obtainable if one does not represent the unfolded matrix in terms of frontal slices of $\underline{\mathbf{X}}$. By predefining matrices with different slice orientations, one can write the alternative matricizations as

$$\mathbf{X}_{\mathrm{b}} = [\mathbf{U}_1'|\mathbf{U}_2'|\cdots|\mathbf{U}_I'] \tag{12}$$

where $\mathbf{U}_i$ is a $K \times J$ horizontal slice of $\underline{\mathbf{X}}$, and

$$\mathbf{X}_{\mathrm{c}} = [\mathbf{U}_1'|\mathbf{U}_2'|\cdots|\mathbf{U}_J'] \tag{13}$$

where $\mathbf{U}_j$ is an $I \times K$ vertical slice of $\underline{\mathbf{X}}$. We see that use of Kiers' subscript actually specifies two things: (i) explicitly, what the row mode is, and (ii) implicitly, that the other modes follow the shorthand's order convention.

Using Kiers' notation on the left and Bro's on the right,

Table II. Matrix representations of eight models using both 'slice' and 'stretch' approaches

	Model	Slicewise representation	Stretched (unfolded) representation
1	Tucker3	Not possible[a]	$\mathbf{X}_{\mathrm{a}} = \mathbf{A}\mathbf{G}_{\mathrm{a}}(\mathbf{C}' \otimes \mathbf{B}')$
2	Tucker2	$\mathbf{X}_k = \mathbf{A}\mathbf{G}_k\mathbf{B}'$	$\mathbf{X}_{\mathrm{a}} = \mathbf{A}\mathbf{G}_{\mathrm{a}}(\mathbf{I}_K \otimes \mathbf{B}')$
3	TMMDS/IDIOSCAL	$\mathbf{Y}_k = \mathbf{A}\mathbf{H}_k\mathbf{A}'$	$\mathbf{Y}_{\mathrm{c}} = \mathbf{H}_{\mathrm{c}}(\mathbf{A} \otimes \mathbf{A})'$
		$\mathbf{Y}_k = \mathbf{A}\mathbf{D}_k\mathbf{\Phi}_k\mathbf{D}_k\mathbf{A}'$	$\mathbf{Y}_{\mathrm{c}} = \big((\mathbf{C}' \odot \mathbf{C}')' * \mathbf{\Phi}_{\mathrm{c}}\big)(\mathbf{A} \otimes \mathbf{A})'$
4	Tucker1	$\mathbf{X}_k = \mathbf{A}\mathbf{G}_k$	$\mathbf{X}_{\mathrm{a}} = \mathbf{A}\mathbf{G}_{\mathrm{a}}(\mathbf{I}_K \otimes \mathbf{I}_J)$
5	Parafac1	$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{B}'$	$\mathbf{X}_{\mathrm{a}} = \mathbf{A}\mathbf{I}_{\mathrm{a}}(\mathbf{C}' \otimes \mathbf{B}')$ or $\mathbf{X}_{\mathrm{a}} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})'$
6	Indirect Parafac1 (or orthogonal Parafac2)	$\mathbf{Y}_k = \mathbf{A}\mathbf{D}_k^2\mathbf{A}'$[b]	$\mathbf{Y}_{\mathrm{c}} = (\mathbf{I}_{\mathrm{a}}(\mathbf{C}' \odot \mathbf{C}'))'(\mathbf{I}_{\mathrm{a}}(\mathbf{A}' \otimes \mathbf{A}'))$ or
			$\mathbf{Y}_{\mathrm{c}} = (\mathbf{C}' \odot \mathbf{C}')'\mathrm{diag}(\mathrm{vec}(\mathbf{I}_R))(\mathbf{A} \otimes \mathbf{A})'$
7	Parafac2	$\mathbf{Y}_k = \mathbf{A}\mathbf{D}_k\mathbf{\Phi}\mathbf{D}_k\mathbf{A}'$	$\mathbf{Y}_{\mathrm{c}} = (\mathbf{C}' \odot \mathbf{C}')'\mathrm{diag}(\mathrm{vec}(\mathbf{\Phi}))(\mathbf{A} \otimes \mathbf{A})'$
8	Paratuck2	$\mathbf{X}_k = \mathbf{A}\,{}^{\mathrm{A}}\mathbf{D}_k\mathbf{H}\,{}^{\mathrm{B}}\mathbf{D}_k\mathbf{B}'$	$\mathbf{X}_{\mathrm{c}} = ({}^{\mathrm{B}}\mathbf{C}' \odot {}^{\mathrm{A}}\mathbf{C}')'\mathrm{diag}(\mathrm{vec}(\mathbf{H}))(\mathbf{B} \otimes \mathbf{A})'$

$\mathbf{X}_k$ is a two-way slice (i.e. the *k*th frontal plane) taken from the three-way array $\underline{\mathbf{X}}$; $\mathbf{Y}_k$ (i.e. the *k*th frontal slab of the three-way array $\underline{\mathbf{Y}}$) is equal to $\mathbf{X}_k\mathbf{X}_k'$; $\mathbf{G}_k$ is the *k*th frontal slice taken from the core array **G**; **A, B** and **C** are parameter matrices for modes A, B and C respectively; $\mathbf{D}_k$ is a diagonal matrix holding the *k*th row of **C** in its diagonal; and $\mathbf{\Phi}_k$ is the matrix of cross-products of mode B loadings at the *k*th level of mode C (i.e. the *k*th frontal slice of $\underline{\mathbf{\Phi}}$). In Parafac2 the $\mathbf{\Phi}$ matrix is fixed across all *k*. For the TMMDS/IDIOSCAL model the **H** matrices on the first line are equivalent to the corresponding product in the following line. **H** in Paratuck2 is of course the (possibly rectangular) matrix of interactions between the mode A and B factors.

[a] This becomes possible if one allows a mixture of slice and stretch approaches in a single model. The representative slice can then be written as $\mathbf{X}_k = \mathbf{A}\mathbf{G}_{\mathrm{a}}(\mathbf{c}_k' \otimes \mathbf{B})'$, as pointed out by R. Bro (personal communication, November 2000). It also becomes possible in pure slicewise representation by means of 'encapsulated summation', as explained later in this paper.

[b] This is often written without the exponent when it is clear from the context that the elements are squared quantities.

Equations (11)–(13) are respectively

$$\begin{aligned} \mathbf{X}_{\mathrm{a}} &= \mathbf{X}^{(\mathrm{I}\times\mathrm{JK})} \\ \mathbf{X}_{\mathrm{b}} &= \mathbf{X}^{(\mathrm{J}\times\mathrm{KI})} \\ \mathbf{X}_{\mathrm{c}} &= \mathbf{X}^{(\mathrm{K}\times\mathrm{IJ})} \end{aligned} \tag{14}$$

The shorthand notation is clearly more compact, but the superscript mode-order notation is useful in this example because it allows us to show explicitly how Kiers' mode-order convention is carried out. It makes plain that the first mode appearing in the combination mode is nested in the other one (e.g. mode B is nested in mode C in X_{a}) and that the structures only differ by a cyclic permutation of modes, as stated in Reference [5]. It also shows the size of the combination mode. Note that X_{a}, by itself, does not reveal whether **X** is an unfolded three-, four- or higher-way array, although the context would usually make this clear.

3.4. Khatri–Rao product

The final notational enhancement used to facilitate the 'stretch' approach is the columnwise Kronecker product, typically denoted by '$\odot$' and referred to as the Khatri–Rao product [17].* It can be defined as

$$\begin{aligned} \mathbf{A} \odot \mathbf{B} &\equiv [\mathbf{b}_1 \otimes \mathbf{a}_1 \mid \mathbf{b}_2 \otimes \mathbf{a}_2 \mid \cdots \mid \mathbf{b}_R \otimes \mathbf{a}_R] \\ &\equiv [\mathrm{vec}(\mathbf{a}_1\mathbf{b}_1') \mid \mathrm{vec}(\mathbf{a}_2\mathbf{b}_2') \mid \cdots \mid \mathrm{vec}(\mathbf{a}_R\mathbf{b}_R')] \end{aligned} \tag{15}$$

where **A** and **B** both have the same number of columns and '$\otimes$' is the standard Kronecker product.

By using this product, many models are simplified further. For example, Bro (Reference [6], p. 22) writes matricized Parafac/Candecomp as

$$\mathbf{X}^{(\mathrm{I}\times\mathrm{JK})} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})' \tag{16}$$

4. COMPARISON OF THE TWO APPROACHES

Now we may directly compare the 'stretch' (unfolding or matricizing) and 'slice' (representative matrix) methods of representing three-way structure. The eight models in Table I are rewritten in Table II using both 'slice' and 'stretch' approaches.

The stretched representations in Table II follow Kiers' convention in using a non-italic subscript to indicate the elementary mode. This clearly facilitates a more compact representation, particularly with the Tucker models, but one must use extra care when trying to interpret or compare model structures. For example, some matrices with the same elementary mode may differ in the form of the combination mode, however, depending on the model. This occurs here for $\mathbf{G}_{\mathrm{a}}$ which is $R \times ST$, $R \times SK$ and $R \times JK$ respectively for the T3, T2 and T1 models. Note also that the $\mathbf{\Phi}_{\mathrm{c}}$ in the TMMDS/IDIOSCAL model is a $K \times RR$ unfolded version of an $R \times R \times K$ $\underline{\mathbf{\Phi}}$ in which mode C is the elementary mode. The $\mathbf{I}_{\mathrm{a}}$ used for the Parafac1 models (as suggested in Reference [5]) is an $R \times RR$ matricized version of the three-way super-identity array $\underline{\mathbf{I}}$ which contains ones on the super diagonal and zeros elsewhere, it should not be confused with $\mathbf{I}_K$ (or $\mathbf{I}_J$ or $\mathbf{I}_R$) used in the representation of T2 (or T1 or indirect Parafac1), which is an identity matrix

* The usefulness of this operation in multilinear modeling is demonstrated by the fact that it was independently (re)invented by Bro to facilitate the work described in Reference [6]. For this reason, we have sometimes referred to it as the Khatri–Rao–Bro (KRB) product.

of order K (or J or R respectively). The '*' used in the three-mode scaling model represents the Hadamard (or elementwise) product.

The stretched versions of the indirect fit models (3, 6 and 7) in Table II have been presented in a way that emphasizes their symmetry. However, there is an alternative representation for each that more closely parallels the corresponding direct fit model. For example, rewriting the first IDIOSCAL equation as $\mathbf{Y}_a = \mathbf{AH}_a\ (\mathbf{I}_K \otimes \mathbf{A}')$ (where $\mathbf{H}_a$ includes squared $\mathbf{C}$-weights) puts it in a form more similar to the T2 representation in the table, and rewriting indirect fit Parafac1 as $\mathbf{Y}_a = \mathbf{AI}_a\ (\tilde{\mathbf{C}}' \otimes \mathbf{A}')$, where $\tilde{\mathbf{C}} = \mathbf{C}{*}\mathbf{C}$, stresses its relationship with direct fit Parafac1.

Note how the slicewise representation requires the *ad hoc* construction of a set of diagonal matrices $\mathbf{D}_k$ that are not part of the original parameter sets (in this case, **A, B** and **C**). It also hides the fact that a set of model parameters for mode C is contained in a matrix **C** that is completely parallel in form and role to the parameter matrices **A** and **B**. An additional limitation of this approach is apparent: it is not fully general (unless augmented, as discussed below). It cannot represent the T3 model, because that structure is not 'slicewise regular', since at each level of a given mode (e.g. at each k) a different weighted combination of core array slices is formed to create the matrix relating the other two modes.

However, the slicewise representation is frequently more compact and more intuitively transparent than the 'stretched array' version. For example, the Parafac2 model (Table II, row 7) has a simple intuitive structure in the slicewise representation; it shows that the nature of the variation across levels of mode C is simply a reweighting of the columns of **A**, which is a helpful insight because such reweighting has a simple interpretation as an increase or decrease in factor influence. The relation of Parafac2 to other models is also easily seen. Orthogonal Parafac2 is a special case of Parafac2 where $\mathbf{\Phi} = \mathbf{I}$; in the other direction, it is apparent that Parafac2 is a special case of Tucker's three-mode scaling [4] or Carroll's IDIOSCAL [10,11] (see also J. D. Carroll and J. J. Chang, paper presented at the meeting of the Psychometric Society, Princeton, NJ, March 1972) in which $\mathbf{\Phi} = \mathbf{\Phi}_K$ (i.e. factor interactions are no longer fixed). The same information is implicit in the unfolded representation, but it is less accessible.

Likewise, the slice representation of TMMDS (row 3) is easily modified by decomposing $\mathbf{H}_k$ into an angle matrix and a weight matrix (lower row 3); both Tucker and Carroll do this, because in this form it can be interpreted as a weighted cosine matrix of angles among dimensions. Its unfolded version seems less straightforward. Whereas the slicewise version uses only the standard matrix product, the stretched version also uses the Kronecker, Khatri–Rao and Hadamard products.

4.1. Completeness of slicewise representation

One might initially wonder whether a representative slice (or vector, as below) is a 'complete' mathematical description of an array, but in fact it contains all the information about the array that is present in the unfolded version.

One might also wonder how easy it is to apply mathematical operations and to derive new results when using representative slice notation, but, so long as both sides of each equation are array slices, there is no problem. Sometimes it may even be preferred over stretch notation. For example, even after having developed the matricized approach, Tucker used slicewise notation in many places in his paper on three-mode MDS [4], presumably because it was more effective in that context. The mathematical flexibility and power of slicewise representation are also demonstrated both by its use in the mathematical uniqueness proofs for Parafac/Candecomp [18] and Partuck2 [9] and by its use in the derivation of Parafac2 as a generalization of Parafac1 [19], which we show here as an example.

If we substitute the Parafac1 representation

$$\mathbf{X}_k = \mathbf{AD}_k\mathbf{B}' \tag{17}$$

into the slice representation for covariances (or cross-products if $\mathbf{X}_k$, is not suitably centred and standardised), the Parafac2 model may then be easily obtained as

$$\begin{aligned}\mathrm{Cov}_k &= \mathbf{X}_k\mathbf{X}_k' \\ &= (\mathbf{AD}_k\mathbf{B}')(\mathbf{B}'\mathbf{DA}') \\ &= \mathbf{AD}_k(\mathbf{B}'\mathbf{B})\mathbf{D}_k\mathbf{A}' \\ &= \mathbf{AD}_k\mathbf{\Phi}\mathbf{D}_k\mathbf{A}' \end{aligned} \tag{18}$$

In Reference [3], Tucker does a related derivation using a matricized approach, but it is somewhat less compact.

4.2. Vector-based variants

As a stretching approach, vectorization of multiway data is sometimes useful. Kiers [5] shows the Tucker3 model in vectorized form as

$$\mathrm{vec}(\mathbf{x}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A})\mathrm{vec}(\mathbf{g}) \tag{19}$$

and the Parafac1 model as

$$\mathrm{vec}(\mathbf{x}) = (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})\mathbf{1}_R \tag{20}$$

in which $\mathbf{1}_R$ is an order R vector of all ones. Clearly, the vectorized representation is more symmetrical than the matricized equivalents (models 1 and 5 respectively in Table II), in that the vectorized representation does not need to compound two modes (or N-1 for N-way models) into one combination mode. For these and other models in which all modes are symmetrical, however, both the vectorized and matricized approaches are similar in terms of compactness and ease of manipulation.

Conversely, it is not yet clear to us how to represent some of the models (e.g. non-symmetrical models 3, 7 and 8 in Table II) in vectorized form. Moreover, even if there is a vectorized representation, it would very likely be less transparent than the slice representation.

5. PROPOSED IMPROVEMENTS FOR SLICEWISE REPRESENTATION

We now propose two notational modifications to make slice representation more useful. The first is 'encapsulated summation', which eliminates the need to predefine other matrices and makes more transparent the components of a matrix. The second is 'angle bracket' notation, variations of which have already been used by others. It also improves transparency by eliminating the need to predefine diagonal matrices.

5.1. Encapsulated summation

We have seen Tucker's slicewise representation of the T3 model in (3), where $\mathbf{N}_k$ hides the presence of $\mathbf{C}$ and $\underline{\mathbf{G}}$. If $h_{rsk} = \sum\limits_t c_{kt} g_{rst}$ (cf. Equation 2), we can write

$$\mathbf{H}_k = \left(\sum_{t=1}^{T} c_{kt} \mathbf{G}_t \right) \tag{21}$$

(using $\mathbf{H}_k$ instead of $\mathbf{N}_k$ to be consistent with the Table II notation). We see that $\mathbf{H}_k$ is a sum of weighted $R \times S$ slices of $\underline{\mathbf{G}}$. Using the matrix expression for $\mathbf{H}_k$ instead of $\mathbf{N}_k$ in (3), we can instead represent the T3 model as

$$\mathbf{X}_k = \mathbf{A} \left(\sum_t c_{kt} \mathbf{G}_t \right) \mathbf{B}' \tag{22}$$

We call use of this expression for $\mathbf{H}_k$, instead of $\mathbf{H}_k$ itself, 'encapsulated summation'. A similar technique may be used for other models and with other types of encapsulated expressions.

This representation for T3 is not new (see e.g. Reference [20], p. 184), but it has not been used very often. However, its value can be seen in this context. Encapsulated summation means both that we need not predefine or use $\mathbf{H}_k$ (or $\mathbf{N}_k$ in Tucker's notation) and that the presence of parameters from $\mathbf{C}$ and $\underline{\mathbf{G}}$ is not obscured.

This representation of T3 is not only referentially but also conceptually more transparent. The role of the parameter sets is apparent in the sense that one can easily grasp how each part is combined with the effect of the others. It also plainly shows that the changes in factor/component interactions are restricted to those that can be generated by weighted combinations of slices of the core array $\underline{\mathbf{G}}$, thus clarifying how T3 can be understood as a special case of T2. This is useful, because one important interpretation of T3 is in terms of an 'oblique' factor/component model in which not only the factor sizes or weights but also the relations between the factors change in a restricted way from one level to the next. In spite of these advantages of encapsulated summation for representing T3, however, it still has a limitation: it does not make it immediately apparent that the role of the three modes is entirely symmetrical.

5.2. Angle bracket notation

Instead of transforming vectors into diagonal accessory matrices (e.g. $\mathbf{c}_k$ into $\mathbf{D}_k$), it is simpler and more transparent to use the notation for the original vector and surround it with angle brackets. The convention is that, for any vector $\mathbf{v}$,

$$\langle \mathbf{v} \rangle \equiv \begin{bmatrix} v_1 & 0 & \cdots \\ 0 & v_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{23}$$

The usual notation for this is 'diag($\mathbf{v}$)', but this does not lend itself well to inclusion in larger matrix expressions.

Just as with 'diag( )', when the notation is applied to a matrix, it is interpreted as a vector formed from the diagonal elements. For example, if $\mathbf{P}$ is a matrix with N levels in its smaller mode, then

$$\langle \mathbf{P} \rangle \equiv \begin{bmatrix} p_{11} \\ p_{22} \\ \vdots \\ p_{NN} \end{bmatrix} \tag{24}$$

The first use of angle bracket notation is in References [21] and [22]. Others have probably used related techniques in other contexts. We recently discovered one example where Sidiropoulos *et al.* [23] define a function that generates a diagonal matrix from a designated row of a Parafac factor loading matrix: they say 'let $\mathbf{D}_p$ $(\boldsymbol{\Phi})$ denote the diagonal matrix containing the $(p+1)$-st row of $\boldsymbol{\Phi}$ [then] … $\mathbf{X}_p = \mathbf{A}\mathbf{D}_p$ $(\boldsymbol{\Phi})\mathbf{S}$ ' (p. 2379). This, and probably other similar ideas that were independently developed, achieve similar results; we simply offer the angle bracket notation as one possible alternative, one that might or might not be found to be slightly simpler and slightly more general.

Examples of several different uses of angle bracket notation are presented below.

5.2.1. *Angle bracket notation applied to the Parafac model*

Using angle bracket notation, Equation 17 becomes

$$\mathbf{X}_k = \mathbf{A} \langle \mathbf{c}_k \rangle \mathbf{B}' \tag{25}$$

from which it is immediately apparent that the weights for the kth level of the third mode come from the kth row of $\mathbf{C}$. Generalizations are simple and natural, since n-way Parafac becomes $\mathbf{X}_{klm\cdots} = \mathbf{A} \langle \mathbf{c}_k \rangle \langle \mathbf{d}_l \rangle \langle \mathbf{e}_m \rangle \cdots \mathbf{B}'$.

The notation can be applied in other ways to emphasize different perspectives on the model. For example, the following is a visually evocative or mnemonic form that avoids summation notation:

$$x_{ijk} = \mathbf{1}' \langle \mathbf{a}_i \rangle \langle \mathbf{b}_j \rangle \langle \mathbf{c}_k \rangle \mathbf{1} \tag{26}$$

Once again, the generalization to $x_{ijkl\cdots} = \mathbf{1}' \langle \mathbf{a}_i \rangle \langle \mathbf{b}_j \rangle \langle \mathbf{c}_k \rangle \langle \mathbf{d}_l \rangle \cdots \mathbf{1}$ for the n-way case maintains the same simplicity and symmetry. Angle bracket notation thus provides another way to highlight the symmetry and multilinearity of the model. Its flexibility is also demonstrated by the ease with which an intermediate formulation can be obtained to represent the model in terms of a vector from the array rather than a scalar, as in

$$\mathbf{x}_{jk} = \mathbf{A} \langle \mathbf{b}_j \rangle \langle \mathbf{c}_k \rangle \mathbf{1} \tag{27}$$

5.2.2. *Angle bracket notation applied to other models*

Angle bracket notation also adds transparency to the expression of quasi-multilinear models such as Parafac2 [19,20], where we have

$$\mathbf{Y}_k = \mathbf{A} \langle \mathbf{c}_k \rangle \boldsymbol{\Phi} \langle \mathbf{c}_k \rangle \mathbf{A}' \tag{28}$$

We call this model *quasi*-multilinear because it is, strictly speaking, non-linear (quadratic) in $\mathbf{C}$ and $\mathbf{A}$, since the parameters in these matrices enter twice. For comparison, the fully multilinear analog of (28) is Paratuck2 [8,20], which is written as $\mathbf{X}_k = \mathbf{A}\ \langle {}^{\mathrm{A}}\mathbf{c}_k \rangle\ \mathbf{H}\ \langle {}^{\mathrm{B}}\mathbf{c}_k \rangle\ \mathbf{B}'$. Here there is a separate weight matrix ${}^{\mathrm{A}}\mathbf{C}$ for the mode A factors and ${}^{\mathrm{B}}\mathbf{C}$ for the mode B factors.

Finally, consider the more specialized model called one-mode (singly dependent) PARALIND [24]. It can be made more transparent by writing an expression for a single element as

$$x_{ijk} = \mathbf{1}'\langle \mathbf{a}_i \rangle \mathbf{H} \langle \mathbf{b}_j \rangle \langle \mathbf{c}_k \rangle \mathbf{1}$$

where $\mathbf{H}$ is rectangular because $\langle \mathbf{a}_i \rangle$ and $\langle \mathbf{b}_j \rangle$ have different lengths. It is (typically) fixed in advance to have some special form describing the linear dependences involved in the relationship between factors in different modes.

6. ALTERNATIVES TO MATRIX NOTATION

If one is willing to abandon matrix notation, all the complications discussed here can be avoided. Harshman [25] has recently proposed an alternative to matrix notation which represents arrays directly by incorporating index symbols into their names and by providing rules for operating on them based on the index symbols. For example, x_{IJK} or $\underline{\mathbf{X}}_{\mathrm{IJK}}$ are names of a three-way array. One advantage of this notation is that it eliminates the need for either slicing or stretching, since the three- (or *n*-) way arrays are directly represented and operated on.

For example, the T3 model for a full array is written as $\underline{\mathbf{X}}_{\mathrm{IJK}} = \mathbf{A}_{\mathrm{IR}}\mathbf{B}_{\mathrm{JS}}\mathbf{C}_{\mathrm{KT}}\underline{\mathbf{G}}_{\mathrm{RST}}$ and the T2 model as $\underline{\mathbf{X}}_{\mathrm{IJK}} = \mathbf{A}_{\mathrm{IR}}\ \mathbf{B}_{\mathrm{JS}}\underline{\mathbf{G}}_{\mathrm{RSK}}$. Symmetric T2 is written as $\underline{\mathbf{Y}}_{\mathrm{II'K}} = \mathbf{A}_{\mathrm{IR}}\ \underline{\mathbf{H}}_{\mathrm{RR'K}}\ \mathbf{A}_{\mathrm{I'R}}$ and the TMMDS/IDIOSCAL version that displays the structure of $\mathbf{H}_{\mathrm{k}}$ is $\underline{\mathbf{Y}}_{\mathrm{II'K}} = \mathbf{A}_{\mathrm{IR}}\ (\mathbf{c}_{\mathrm{kr}}\ \boldsymbol{\phi}_{\mathrm{rr'k}}\ \mathbf{c}_{\mathrm{kr'}})_{\mathrm{RR'K}}\ \mathbf{A}_{\mathrm{I'R}}$. The Parafac2 special case is obtained by dropping the 'k' subscript from the $\boldsymbol{\phi}$-element. Note that these expressions do not describe slices or unfolded matrices, but rather the three-way matrices themselves. Similar expressions can be used to represent four- or *n*-way arrays.

Another alternative has been proposed by Alsberg [26] and further studied and extended by D. S. Burdick (paper presented at TRICAP 2000, the Third Annual Meeting on Three-way Methods in Chemistry and Psychology, Faaborg, July 2000). It is called 'ball and whisker' notation and is based on graphical objects and rules for their connection and transformation. It can express in *visual* terms the operations necessary for *n*-way multilinear algebra.

7. CONCLUSION

When choosing a matrix method for expressing the latent structure of a particular three-way array and/or representing transformations of its components, it is often useful to consider both the nature of the model and one's objectives. With respect to models, stretched array or matricized notation is well suited to some models such as T3, but slicewise notation more concisely expresses other models such as Parafac2. In terms of objectives, stretched array notation (matricized or vectorized) is particularly useful for the description and development of ALS estimation algorithms. Sliced array representation is often useful for compact and accessible communication of the ideas behind a given model in terms of its functional units and their relations.

Just as the Bro and Kiers conventions for unfolded array notation produce improvements in the stretched array approach, so too the proposed angle bracket notation, and for certain models the encapsulated summation notation, will provide improvements in the representative slice approach.

Acknowledgements

The authors wish to thank Henk Kiers for helpful comments on the matricizing approach, Rasmus Bro for useful discussions on points addressed in the paper, and our two reviewers for the precision and care with which they went over the manuscript. The first author was supported by Natural Sciences and Engineering Research Council of Canada research grant OGP-000-7896.

REFERENCES

1. Tucker LR. Implications of factor analysis of three-way matrices for measurement of change. In *Problems in Measuring Change*, Harris CW (ed.). University of Wisconsin Press: Madison, WI, 1963; 122–137.
2. Tucker LR. The extension of factor analysis to three-dimensional matrices. In *Contributions to Mathematical Psychology*, Gulliksen H, Frederiksen N (eds). Holt, Rinehart and Winston: New York, 1964; 110–127.
3. Tucker LR. Some mathematical notes on three-mode factor analysis. *Psychometrika* 1966; **31**: 279–311.
4. Tucker LR. Relations between multidimensional scaling and three-mode factor analysis. *Psychometrika* 1972; **37**: 3–27.
5. Kiers HAL. Towards a standardized notation and terminology in multiway analysis. *J. Chemometrics* 2000; **14**: 105–122.
6. Bro R. *Multi-way Analysis in the Food Industry: Models, Algorithms and Applications*. University of Amsterdam: Amsterdam, 1998; http://www.mli.kvl.dk/staff/foodtech/brothesis.pdf [1 March 2001].
7. Kroonenberg PM. *Three-mode Principal Component Analysis: Theory and Applications*. DSWO Press: Leiden, 1983.
8. Harshman RA and Lundy ME. PARAFAC: parallel factor analysis. *Comput. Statist. Data Anal.* 1994; **18**: 39–72.
9. Harshman RA and Lundy ME. Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP. *Psychometrika* 1996; **61**: 133–154.
10. Carroll JD and Arabie P. Multidimensional scaling. *Ann. Rev. Psychol.* 1980; **31**: 607–649.
11. Carroll JD and Wish M. Models and methods for three-way multidimensional scaling. In *Contemporary Developments in Mathematical Psychology, vol. 2*, Krantz DH, Atkinson RC, Luce RD, Suppes P (eds). Freeman: San Francisco, 1974; 57–105.
12. Carroll JD and Chang JJ. Analysis of individual differences in multidimensional scaling via an *N*-way generalization of Eckart–Young decomposition. *Psychometrika* 1970; **35**: 283–319.
13. Harshman RA. Foundations of the PARAFAC procedure: models and conditions for an 'explanatory' multimodal factor analysis. *UCLA Working Papers Phonet.* 1970; **16**: 1–84; http://publish.uwo.ca/~harshman.
14. MacDuffee CC. *The Theory of Matrices*. Chelsea Publishing: New York, 1946.
15. Steeb W-H. *Matrix Calculus and Kronecker Product with*

Applications and C^{++} Programs. World Scientific Publishing: London, 1997.

16. Winer BJ. *Statistical Principles in Experimental Design* (2nd edn). McGraw-Hill: New York, 1971.
17. Khatri CG and Rao CR. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhya Ser. A* 1968; **30**: 167–180.
18. Harshman R. Determination and proof of minimum uniqueness conditions for Parafac1. *UCLA Working Papers Phonet.* 1972; **22**: 111–117; http://publish.uwo.ca/~harshman.
19. Harshman R. Parafac2: mathematical and technical notes. *UCLA Working Papers Phonet*. 1972; **22**: 30–44; http://publish.uwo.ca/~harshman.
20. Harshman RA and Lundy ME. The PARAFAC model for three-way factor analysis and multidimensional scaling. In *Research Methods for Multimode Data Analysis*, Law HG, Snyder Jr CW, Hattie JA, McDonald RP (eds). Praeger: New York, 1984; 122–215; http://publish.uwo.ca/~harshman.
21. Harshman RA, Hong S and Lundy ME. Shifted factor analysis. I. Models and properties. *J. Chemometrics*, submitted, under review.
22. Hong S and Harshman RA. Shifted factor analysis. III. N-way generalization and application. *J. Chemometrics*, submitted, under review.
23. Sidiropoulos ND, Bro R and Giannakis GB. Parallel factor analysis in sensor array processing. *IEEE Trans. Signal Process.* 2000; **48**: 2377–2388.
24. Bro R, Harshman RA and Sidiropoulos ND. Rank-deficient models for multiway data. I. Theory. *J. Chemometrics*, submitted, under revision.
25. Harshman RA. An index formalism that generalizes the capabilities of matrix notation and algebra to ***n***-way arrays. *J. Chemometrics* 2001; **15**: 689–714.
26. Alsberg BK. A diagram notation for ***N***-mode array equations. *J. Chemometrics* 1997; **11**: 251–266.