JOURNAL OF CHEMOMETRICS
*J. Chemometrics* 2003; **17**: 363–378
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cem.808

# Shifted factor analysis—Part I: Models and properties

**Richard A. Harshman*, Sungjin Hong and Margaret E. Lundy**
University of Western Ontario, London, Ontario N6A 5C2, Canada

Received 20 July 2001; Revised 31 March 2003; Accepted 16 May 2003

**The factor model is modified to deal with the problem of factor shifts. This problem arises with sequential data (e.g. time series, spectra, digitized images) if the profiles of the latent factors shift position up or down the sequence of measurements: such shifts disturb multilinearity and so standard factor/component models no longer apply. To deal with this, we modify the model(s) to include explicit mathematical representation of any factor shifts present in a data set; in this way the model can both adjust for the shifts and describe/recover their patterns. Shifted factor versions of both two- and three (or higher)-way factor models are developed. The results of applying them to synthetic data support the theoretical argument that these models have stronger uniqueness properties; they can provide unique solutions in both two-way and three-way cases where equivalent non-shifted versions are under-identified. For uniqueness to hold, however, the factors must shift independently; two or more factors that show the same pattern of shifts will not be uniquely resolved if not already uniquely determined. Another important restriction is that the models, in their current form, do not work well when the shifts are accompanied by substantial changes in factor profile shape. Three-way factor models such as Parafac, and shifted factor models such as described here, may be just two of many ways that factor analysis can incorporate additional information to make the parameters identifiable. Copyright © 2003 John Wiley & Sons, Ltd.**

**KEYWORDS:** factor shifts; sequential data; unique solutions; time series; spectral shifts; three-way or three-mode analysis; Parafac; Parafac2; Tucker T3 and T2; principal components; latent variables; latent structure models

## 1. INTRODUCTION

### 1.1. Uniqueness and the 'rotation' problem

Factor (or principal component) analysis is sometimes used to recover the pattern contributed by each individual source from a mixture of patterns contributed by several sources. In chemistry, each pattern might be a spectrum of light emission energies and each source might be a distinct compound; in psychology, each pattern might be a set of opinions likely to be held and each source might be a distinct personality trait, or each pattern might be a set of test problems likely to be solved and each source might be a specific cognitive ability. (For a general discussion see e.g. References [1,2].) However, correct recovery of the source patterns is hampered by the fact that the bilinear model used in factor/component analysis is 'under-identified'; it does not provide a single unique solution but only a family of alternatives with equally good fit, corresponding to a family of axis 'rotations' in the space spanned by the factors.

*Correspondence to: R. A. Harshman, Psychology Department, 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.

Considerable effort has gone into the search for methods to identify the most valid member of the family of possible solutions. In chemistry, approaches to the problem have included, for example, evolving factor analysis [3], rank annihilation factor analysis [4,5], local rank [6] and alternating regression [7,8]. Often these methods have incorporated constraints such as non-negativity [9] or 'selectivity' [10]. See Reference [11] (pp. 135–184) for a discussion of these and other types of constraints that may be used. In psychology, approaches have included development of criterion rotation [12] or rotation to theoretical targets (see e.g. Reference [13] and Reference [14], pp. 353–360). Other rotation methods optimize some property of the solution, most frequently some measure of 'simplicity' (e.g. Varimax; see Reference [15], pp. 422–441, and Reference [16]). General principles of factor rotation are presented in Reference [17] and Reference [18] (pp. 176–182, 231–238).

A different approach is to modify the factor model itself so that it incorporates more information and describes more structure and, as a consequence, is no longer under-identified. To date, this has primarily been done by generalizing the bilinear factor model, which fits a single two-way data matrix, into a three-way model that simultaneously fits several related data matrices at once [11,19,20]. The simplest such model is trilinear (or higher-way multilinear) and,

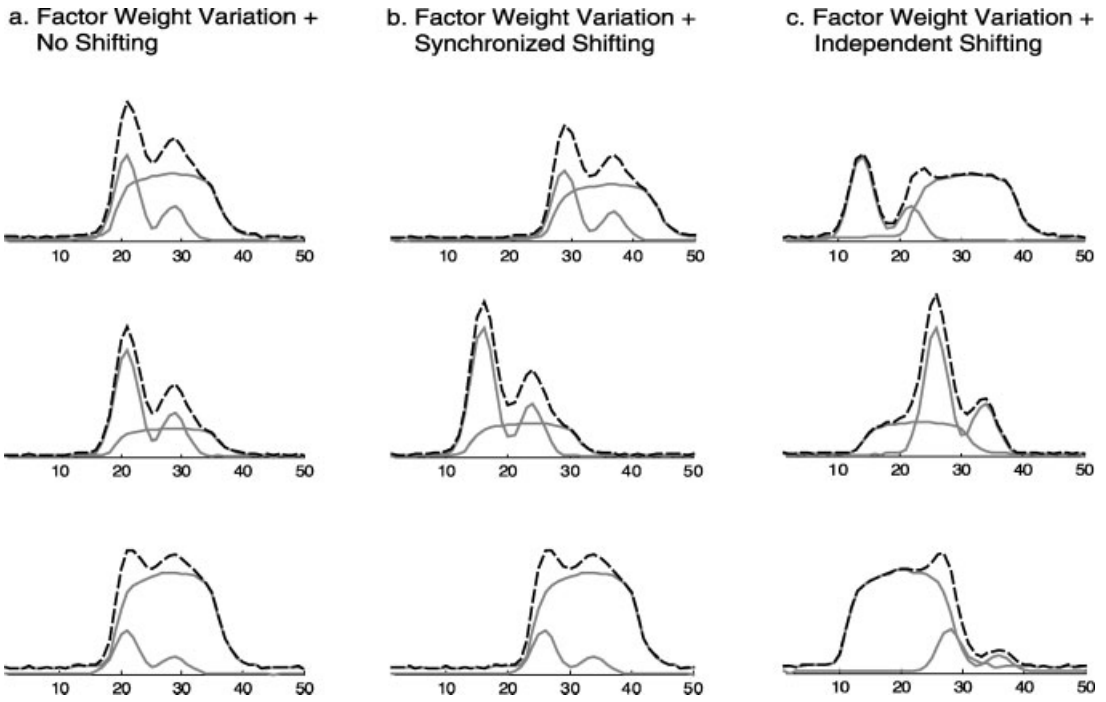


**Figure 1.** Two kinds of factor shifts and their effects. Full and broken lines indicate factor and data profiles respectively.

under conditions that in some situations are relatively natural, the best fitting factor solution is unique except for trivial variations that do not affect interpretation (e.g. changing order of factors) (see e.g. References [21,22]).

In this paper, however, we examine a *different* way to modify or strengthen the factor model so that it is no longer under-identified. Here the added information incorporated into the model, as well as the added structure described by the model, concerns the independent 'shifting of position' of factors in a sequential data set. Inability to deal with such shifts becomes an important limitation of current multilinear methods when they are applied to some types of sequential data.

## 1.2. Shifted factors in sequential data

For certain kinds of data, the order or relative position of the measurements carries important meaning. The prime examples of this are time series data, spectral data and spatial (e.g. image or map) data. We will refer to such data as *sequential**. If a vector $\mathbf{v}$ contains sequential values, then elements with adjacent subscripts, such as $v_1$, $v_2$ and $v_3$, represent measurements taken at adjacent points along some measurement continuum. If, for example, the continuum is time, then $v_2$ is later in time than $v_1$ and earlier in time than $v_3$.

Factor analysis of such data does not, in itself, pose any problem. The extra meaning implicit in the ordering of the measurements is preserved or conveyed in the ordering of the loadings. Observed sequential profiles are modeled as weighted linear combinations of latent sequential profiles.

New principles are required, however, when the variation in latent sequential profiles goes beyond simple weighting. In this paper we consider the case where the overall position of the profile (i.e. of its particular peaks, valleys, etc.) can *shift* up or down on the measurement sequence (see Figure 1). By 'shift' we simply mean that the values making up the profile systematically change subscript location: if $\tilde{\mathbf{v}}$ is a shifted version of $\mathbf{v}$, then $\tilde{v}_i = v_{i+s}$, where $s$ represents the amount of shift. In the shifted factor generalization, observed sequential profiles are modeled as linear combinations of weighted and possibly *shifted* latent sequential profiles.

Such latent shifts can arise in many areas of science. Shifting of *spectral profiles* can occur, for example, in astronomy when the signal sources are in motion relative to the observer, as in the well known red shift of light from distant galaxies, and the complex mixtures of shifted spectra that arise from the violent activity near some black holes. In chemistry and physics a wide variety of absorption and emission spectra are subject to problems of shifting position (see below). Shifting of *time series* can occur, for example, in acoustic or electromagnetic signal processing when mixtures of similar signals arising from widely separated sources are recorded using an array of sensors (microphones, antennae, etc.) located in different places. The differences in relative distance between each source and each sensor produces a pattern of shift in the arrival times of the underlying time series (signal waveforms) from one sensor location to another*. The pattern changes in an orderly way if the sources and/or sensors change position from one sample of recordings to the next, potentially generating three-way or higher-order multiway data sets. Shifting of *spatial patterns* can occur when several images are composites of the same underlying spatial patterns but these patterns are displaced in the image plane (or space) by amounts that vary from one image to the next. This occurs, for example, in functional brain imaging (fMRI) and in binocular vision.

*In the case of images or maps the 'sequences' are orderings in two or more dimensions. Nonetheless, since the added information still consists of adjacency and distance relations, we also refer to these data as 'sequential'.

*For a Parafac analysis of such data *without* the shifted model proposed here, see Reference [23].

### 1.2.1. *Examples from chemometrics and in neuropsychology*

1.2.1.1. Chemometrics. The problem of spectrum position shifts is widespread in chemometrics and arises in several different ways[†]. These include: (a) all hyphenated methods in which chromatography is hyphenated with spectroscopy (e.g. LC/UV, LC/NIR, GC/IR, GC/MS); (b) on-line spectroscopic methods where external variation is present, such as temperature or viscosity changes (mostly in NIR, Raman spectra); (c) instrument drift over the long term, which sometimes causes shifts (mainly IR, NIR, NMR); and (d) differences between two instruments of the same brand (e.g. two NIR instruments should give the same spectra for a compound, and if they do not, shifts may occur).

Not all shifting is equally complicated, however. Tauler [24] points out the important distinction between the case where shifts are the same for all factors and the case where shifts of one factor differ from those of another. Extending his terminology, we call the first case 'synchronized' factor shifting (a.k.a. measurement level shifting; column b in Figure 1) and the second 'independent' factor shifting (a.k.a. latent level shifting; column c in Figure 1). Tauler proposes specific multilinear analysis strategies to deal with each and points out that trilinear methods are not appropriate for the shifted cases, particularly when shifts are not synchronized.

Bro *et al.* [25] considered the problem of shifts in the context of analyzing three (and four)-way data from chromatography. They showed how shifts and other changes in the elution profiles degraded the results returned by a straightforward Parafac analysis. To solve the shift problem, they directly fit Parafac2 to these data, which gave better results. In Part III [26] we reanalyze their data and compare the advantages and limitations of their approach with those of shifted factor analysis (SFA).

1.2.1.2 Neuropsychology. Field and Graupe [27] encountered the problem of factor shifts in an analysis of brain electrical potentials elicited by light flashes. They did a Parafac analysis of a time (after stimulus) by subject by electrode location data array and convincingly demonstrated that factor shifts (i.e. variation in arrival time of the signals generated by underlying brain events, due to changes in subject excitability or other influences) resulted in the creation of an artificial additional factor. The factor was shaped like the first derivative (in time) of the 'jittering' component of the electrical evoked response, and it had subject weights that correlated 0.77 with subject latency variations in the data.

### 1.2.2. *Added complications due to shape changes*

Often it is not only a shift of peak position that occurs but also a change in peak shape. For example, see Reference [28] and references cited therein. Wulfert *et al.* [28] studied the effects of shifts and shape changes on the fit and predictive ability of multivariate models, taking as their example short-wave NIR spectra of ethanol/water/isopropanol mixtures at different temperatures. They showed that 'Spectra that exhibit shifts or other changes in their shape do not conform to the linearity demand and consequently a multivariate model will have to use more regression factors than is to be expected by the chemical rank (number of components in the mixture)' (p. 1762).

Provision for peak shape change is not incorporated into the models to be described in this paper. However, many of our results (including uniqueness results) seem to be extendible to shape change situations, as will be explained in Section 6.

## 1.3. Previous work on the shift problem

To deal with the problem of alignment of time series, Cattell [29] proposed his 'time-corrected P-technique'. This method attempts to correct a mismatch of position before the analysis begins, allowing one to subsequently factor the variables in the standard way. To accomplish this, the variables were shifted relative to one another by amounts that maximized cross-variable correlations (considered pairwise), shifting to maximize the correlation for each pair, repeating for all possible pairs and then iterating the whole process until a globally maximum sum (neglecting sign) of all pairwise correlations was obtained. Although this method permitted one variable to be lagged relative to another, it did not consider the possibility of individual factors within the same variable having different lags.

Molenaar [30] overcame this limitation and generalized Cattell's technique by allowing differing lags of individual factors. The method was called dynamic factor analysis (DFA). It was constructed to analyze time series data, but the parameters were estimated by fitting the covariance matrix that such time series would produce. Because of its greater flexibility and additional parameters, the model was considerably more complicated than Cattell's approach and, in fact, more complex than the SFA models to be considered here. We refer the reader to Reference [30] for further details.

Both SFA and DFA incorporate shift (or time lag) variation as a part of the model's latent variable parameters. In contrast, there have recently been some new data-preprocessing techniques proposed for peak realignment that are (like Cattell's method) applied directly to the 'surface' data. These are, for example, proposed for chemometric applications such as analysis of retention time-shifted chromatographic spectra or frequency-shifted spectroscopic profiles. Some of these techniques are described in the following paragraphs.

Prazen *et al.* [31] considered the retention time shift problem in second-order chromatographic/spectroscopic data. They took advantage of the fact that position shifts increase the complexity of the data structure. They first adjoined a two-way standard LC/UV data matrix (where underlying spectra were lined up) with a sample LC/UV data matrix (where underlying spectra were subject to unknown shifts) so that the two matrices shared the same time axis. Were it not for the position shifts in the sample data, the augmented data would have had the same 'essential' data rank (i.e. ignoring noise) as the standard data. This should also match the number of chemical components. Based on this, the method considers the variance contributed by the first $R$ eigenvalues, where $R$ is the 'essential' rank of the standard (or the number of analytes—known *a priori*). It then makes position shifts of the profiles in the sample data so

[†]Our thanks to Dr A. Smilde for his help with this subsection.

that the variance due to the $R+1$ and later eigenvalues is minimized. The type of shift is the same as in Cattell's method, a shift of the full profile for each sample, but the definition of optimum shift uses an external standard rather than internal correlations. Fraga *et al.* [32] used the same method to adjust for retention time shifts in two-dimensional gas chromatography (GC/GC).

Vogels *et al.* [33] developed a method that was more flexible. Their approach compares *portions* of each individual spectrum with the mean spectrum and then shifts each portion left or right until the sum of squared differences between a particular region in a sample spectrum and the corresponding region in the mean spectrum is minimized. They called it the partial linear fit (PLF) algorithm and applied it to position shifts of peaks in NMR spectroscopic data.

Other methods go beyond simple shifting. Nielsen *et al.* [34] used time warping instead of simple shifting as a data-preprocessing tool, applying it to retention time shifts in chromatographic profiles. Time warping adjusts for shape changes such as broadening or narrowing of peaks and, to some extent, for position shifts as well. Bylund *et al.* [35] applied the same approach to the retention time shift problem in LC/MS data profiles in the Parafac-modeling context and appear to have made an additional refinement: an adjustment for overall shifts.

Another position and shape adjustment technique, used by Westad and Martens [36], is gradient-based motion estimation. In this method, speeds of 'flow' of spectral shape, represented by shift vectors, are determined using difference profiles between sample spectra and a reference spectrum. Then the estimated shift vectors are used for alignment of sample spectra. Like the time-warping method, this approach flexibly adjusts for complex shape changes as well as position shifts. (See Reference [37] for a more detailed description of the algorithm.)

## 2. REPRESENTING SHIFTS: THE SHIFT OPERATOR AND ARRAY NOTATION

We will be discussing models that represent factor shifts by means of systematic changes in the subscript values of array elements. Consequently, we will need an efficient way to describe such changes. The only way to do this with current notation (as far as we know) is to write a scalar expression and replace the usual $i$ and $j$ subscripts with more complex expressions that show how subscript values are to be computed, such as $x_{i+s}$. This is not suitable when we need an expression for a full matrix or array incorporating shifted elements (e.g. in order to use it in a matrix product or some other matrix expression) or when we want to take an equation involving shifted vectors or matrices and manipulate it to get a new equation involving the same objects with different amounts of shift.

There are (at least) two ways to deal with this problem: one is to define a new addition to matrix notation—a mathematical operator that performs matrix index shifts; the other is to change one's basic array notation from standard matrix–vector conventions to one in which array index shifts can be represented naturally, without the need for special operators. Both approaches have advantages. On the one hand, use of the matrix operator allows the shifted factor models to be represented in papers and contexts that use conventional matrix notation throughout. On the other hand, the array notation to be used here can represent shifted arrays of all kinds without requiring a new special operator. In some sense, it provides the only complete solution, since it can express shifted factor models that have no straightforward representation in terms of the shift operator (e.g., see section 5, below). However, to facilitate the introduction of these shifted models, and to aid future workers who may want to adopt either approach, we shall lay out our definitions in both ways. In Parts II and III, we will primarily use the shift operator, since it is compatible with the models discussed there and allows us to use matrix notation throughout, thus reducing the learning effort required of the reader.

### 2.1. The shift operator

To facilitate the expression of multilinear models* with shifted components, we define a shift operator (or function) $\mathcal{S}_{\mathbf{s}_j}(\cdot)$ which takes some (or all) values in an array and assigns them new subscript numbers, thus 'shifting' them to new locations in the array. Each subscript in a set of shifted elements is incremented in size by the same fixed amount, which is determined by an argument given to the operator. When applied to a vector of sequential data, this causes the profile of peaks and valleys to shift position by that amount along the sequence.

The two most common (and simplest) applications of the operator will be to shift a vector and to shift the columns of a matrix. This generalizes in a natural manner to three- and $n$-way arrays and to expressions for subarrays. For an explanation of these generalizations and a more complete general discussion of the mathematical properties of the shift operator, see the Appendix. The employment of the operator in subsequent sections of this paper will provide additional examples that demonstrate its use.

### 2.2. Array index notation

An alternative to the shift operator is the use of Array Index Notation (AIN) [38]. Here subscripts occur explicitly not only in the names of array elements but also in the names of the arrays themselves. Individual letters that indicate sets of subscript values can be replaced by expressions that represent a systematically modified set of these subscript values, and hence an array name can directly express a particular shift of its elements.

#### *2.2.1. Two basic rules*

AIN notation is described fully in Reference [38]. For our current purposes we summarize two basic rules governing

*And models which we would call either quasi- or semi-multilinear. Sometimes, when being precise, we follow a convention that uses distinct and specific names for two classes of models with mostly multilinear structure but with certain violations of multilinearity. Models that are multilinear except that some sets of parameters enter more than once (e.g. Parafac2) are referred to as 'quasi-multilinear', because they are quadratic or cubic, etc. in these parameters. Models in which parameter sets for some modes combine in a simple multiplicative or multilinear fashion, but those for others enter or combine in a different way, are referred to as partly multilinear or semi-multilinear. For convenience, these distinctions are often ignored and the term 'multilinear' is used loosely to refer to all these models.

AIN that are important for this paper, and point out some of their implications.

2.2.1.1. Uppercase subscripts. When a name refers to an element in an array, the subscripts in the name are written in lowercase, as in $x_{ij}$. When the name refers to the entire array, the subscripts are written in uppercase, as in $x_{IJ}$; when it refers to a subarray or a part of an array, only subscripts that take on multiple values in the part referred to are written in uppercase, as in $x_{Ij}$, which is the name of a (generic) column of $x_{IJ}$. Uppercase subscript letters denote the *set** of index values for that subscript position, rather than just a single value, and so any name containing that uppercase subscript represents a *set* of elements, in other words, an array or subarray. Some further examples: the vector conventionally written in matrix notation as $\mathbf{v}$ with elements $v_i$ is written in AIN as $v_I$ with elements $v_i$; likewise, the matrix $\mathbf{Y}$ with rows $\mathbf{y}_r$ and columns $\mathbf{y}_s$ and elements $y_{rs}$ is represented in AIN as $y_{RS}$ with rows $y_{rS}$, columns $y_{Rs}$ and elements $y_{rs}$.

2.2.1.2. Summation convention. The AIN system of notation follows a convention (often called the 'Einstein summation convention') which specifies that whenever identical subscript set names occur on two arrays being multiplied together, such as the R in $a_{IR}b_{JR}$, corresponding elements from the two arrays (elements that have matching values for the subscripts with matching names, e.g. $a_{i3}$ and $b_{j3}$) should be pairwise multiplied together and the products summed; in this example the sums have the form $a_{i1}b_{j1} + a_{i2}b_{j2} + \cdots$. The sum is placed into the product array at the location determined by the non-matching subscripts. For example, if $c_{IJ} = a_{IR}b_{JR}$, elements in the product are computed as $c_{ij} = \sum_r a_{ir}b_{jr}$. Hence in this example the expression $a_{IR}b_{JR}$ specifies the standard matrix product $\mathbf{AB}'$. In AIN the same principle is also extended to more than two coefficients with matching subscripts, so that $p_{IJK} = a_{IR}b_{JR}c_{KR}$ means $p_{ijk} = \sum_r a_{ir}b_{jr}c_{kr}$, thus providing another way of writing the Parafac/Candecomp model.

### 2.2.2. *Representation of shifted arrays*

With this notation it becomes a relatively simple matter to represent arrays in which subscripts are shifted in value. For example, the vector $\tilde{\mathbf{v}}$ was defined previously as being the same as vector $\mathbf{v}$ except that all the subscript positions have been increased by an amount $s$. The equation relating the two vectors would be written in matrix notation using the shift operator as $\tilde{\mathbf{v}} = \mathcal{S}_s(\mathbf{v})$, but it can be written in AIN without any special operator as simply $\tilde{v}_I = v_{[I+s]}$. An array that is shifted across levels of a particular mode (e.g. A) by a particular amount (e.g. $s$) can be written as $y_{[I+s]JK}$; in this case, only column vectors are shifted.

If the shift value added to the sum varies across the levels of some mode, then the shift value appearing in the computed subscript is itself subscripted. For example, in the array $y_{I[J+s_k]K}$, the values of the subscript j are shifted by (potentially) different amounts at each different level k of the third mode. It is also common that a shift value depends on the levels of two other modes, as in $y_{I[J+s_{ik}]K}$. These expressions can be written more formally in AIN by using the rules for computed elements. For our current example the expression could be written as $\left(y_{i[j+s_{ik}]k}\right)_{IJK}$. Note that by using a name containing subscripts (in this case $s_{ik}$) to refer to the shift value, such expressions imply the existence of a separate matrix, in this case $s_{IJ}$ (or in matrix notation, $\mathbf{S}$) containing the shift values.

The use of AIN eliminates the need for a special shift operator, but some of the shift-related issues still remain. In particular, the principles adopted for dealing with ends of shifted vectors must be established and agreed on, just as they must with use of the shift operator (see the Appendix).

### 2.2.3. *Further generality possible*

In addition to its flexibility in dealing with one-dimensional sequences, AIN allows the straightforward statement of more general models, such as those involving shifts in the plane (e.g. of an image) or in three-space (e.g. of a brain scan). For example, in the array $y_{[I+t][J+u]K}$, entire slices are shifted: each mode C slice (i.e. each frontal plane of the array) is shifted by $t$ units in row position and by $u$ units in column position.

In certain cases it is natural to have some of the constituent sets of model parameters organized into three-way or higher-way arrays. One example, involving three-way factor loading arrays, will be presented later in this paper. It demonstrates how an interesting and potentially useful shifted factor model can be simple to state in array notation but be difficult or impossible to state in conventional matrix notation.

## 3. THE TWO-WAY SHIFTED FACTOR MODEL

Consider a matrix $\mathbf{X}$ containing sequentially organized data (e.g. a set of time series, spectral or spatial measurements, etc.). Using standard multiway terminology, we say that the two-way array $\mathbf{X}$ has two 'modes' (short for 'modes of classification' [39]). By convention, the row positions are called levels of mode A and are indexed by $i$, while the column positions are called levels of mode B and are indexed by $j$. Two additional conventions are adopted for shifted factor models: (a) We will always arrange the data so that mode A (rows) has the sequentially ordered levels. Thus row positions correspond to positions along the sequence (e.g. successive time points, frequencies, etc.), and each column contains one sequential profile. Different columns usually correspond to different sources or conditions of measurement (e.g. different variables, objects, etc.). (b) By convention, mode B is the one that controls the amount of shifting (i.e. it is the mode in which the amounts of factor shift change from one level to the next). This is summarized by saying that mode A is the 'shift*ed*' mode and mode B is the 'shift*ing*' (or 'shift-controlling') mode.

Of course, so long as there are no factor shifts, we can represent sequential data in standard matrix notation using the standard factor/component model. We write it

*In all the cases considered here, these sets are not just collections of elements, but *ordered* collections.

here as

$$\mathbf{X} = \mathbf{A}\mathbf{B}' \tag{1}$$

where $\mathbf{A}$ is an $I \times R$ matrix in which $a_{ir}$ gives the relative contribution of factor $r$ at sequence location $i$, and hence the columns of $\mathbf{A}$ describe the latent sequential vectors. Likewise, $\mathbf{B}$ is a $J \times R$ matrix in which $b_{jr}$ gives the relative contribution of factor $r$ (i.e. of latent sequential vector $r$) for the $j$th column of $\mathbf{X}$. (Note that here and below we omit the error or residual term of these models and focus on the structural part. The error terms are sometimes included in Part II [40] where fitting of fallible data is discussed.)

### 3.1. Generalization stated in array notation

In AIN the standard unshifted bilinear model is written as

$$x_{\mathrm{IJ}} = a_{\mathrm{IR}} b_{\mathrm{JR}} \tag{2}$$

where $x_{\mathrm{IJ}} = \mathbf{X}$, $a_{\mathrm{IR}} = \mathbf{A}$ and $b_{\mathrm{JR}} = \mathbf{B}$.

In this notation a limited generalization of (2) that allows for synchronized factor shifts can be written as

$$x_{\mathrm{IJ}} = a_{[\mathrm{I}+\mathrm{s_j}]\mathrm{R}} b_{\mathrm{JR}} \tag{3}$$

where the parameter $\mathrm{s_j}$ gives the amount of shift that *all* factors show at level j of the data. This is the 'time-corrected P-technique' model considered by Cattell [29]. As noted earlier, it is equivalent to a model written in terms of shifts of the $j$th data vector, and so could be rewritten as $x_{\mathrm{IJ}} = \tilde{x}_{[\mathrm{I}+\mathrm{s_j}]\,\mathrm{J}}$.

In this paper we focus our interest on a more general model, one that allows for independent shifts of each factor at each level $j$ of the data. It is written in AIN as

$$x_{\mathrm{IJ}} = a_{[\mathrm{I}+\mathrm{s_{jr}}]\mathrm{R}} b_{\mathrm{JR}} \tag{4}$$

Here the shift parameter $\mathrm{s_{jr}}$ gives the shift at column $j$ that is exhibited by factor or component $r$. (In other words, as the subscript set J goes through its range of values, the j on $\mathrm{s_{jr}}$ goes through the same range, and likewise for R and r*.) Equation (4) is the basic two-way version of shifted factor analysis (SFA).

### 3.2. Generalization stated using the shift operator

Unfortunately, the familiar matrix formulation of the unshifted model given in (1) cannot be economically converted to a shifted factor form (as will become clear below). To facilitate generalization of a matrix formulation, we restate the model using standard notation but in a 'representative vector' form (see Reference [41] on slice-based and vector-based notation).

The equivalent of (1) in a 'representative vector' formulation is

$$\mathbf{x}_j = \mathbf{A}\mathbf{b}_j \tag{5}$$

Here $\mathbf{x}_j$ is column $j$ of $\mathbf{X}$, $\mathbf{A}$ is as before and $\mathbf{b}_j$ is an $R \times 1$ column (i.e. row $j$ of $\mathbf{B}$) that gives the relative weights or contributions of the $R$ factors on occasion $j$. Then, by using the shift operator described above (and in the Appendix), we can write a shifted factor equivalent of (5) as

$$\mathbf{x}_j = \sum_{r=1}^{R} \mathop{\mathcal{J}}_{s_{jr}} (\mathbf{a}_r) b_{jr}$$

or, more compactly,

$$\mathbf{x}_j = \mathop{\mathcal{J}}_{\mathbf{s}_j} (\mathbf{A}) \mathbf{b}_j \tag{6}$$

where, as before, $\mathbf{s}_j$ comes from $\mathbf{S}$, a $J \times R$ matrix giving the shift sizes for columns 1 to $J$ for each of the factors 1 to $R$. Replacing the scalar $s_{jr}$ by the vector $\mathbf{s}_j$ allows us to use the shift operator on $\mathbf{A}$ instead of $\mathbf{a}_r$, which yields a more compact expression, where $\mathbf{s}_j$ is a vector of $R$ shift values taken from row $j$ of the (implied) shift matrix $\mathbf{S}$.

Finally, by repeated use of (6) we can construct a shifted factor equivalent to (1) which does describe the shifted factor structure of the full matrix $\mathbf{X}$, but it has the somewhat awkward form

$$\mathbf{X} = \left[ \mathop{\mathcal{J}}_{\mathbf{s}_1} (\mathbf{A}) \mathbf{b}_1 \middle| \mathop{\mathcal{J}}_{\mathbf{s}_2} (\mathbf{A}) \mathbf{b}_2 \middle| \cdots \middle| \mathop{\mathcal{J}}_{\mathbf{s}_J} (\mathbf{A}) \mathbf{b}_J \right] \tag{7}$$

## 4. UNIQUENESS PROPERTIES OF THE TWO-WAY MODEL

Of key interest here is the uniqueness of the solution (i.e. identifiability of the model parameters given a set of data) for the model given by (4) and (6) or (7). We have investigated this both mathematically and empirically. Both lines of investigation provide strong support for the conjecture that incorporation of information on shifts can strengthen the model enough to make the parameters identifiable up to scaling and column order, and hence make them, in Kruskal's terminology, 'essentially unique'. (In this paper we will follow the common practice of dropping the qualifier 'essential'.) This identifiability appears to hold under relatively mild conditions, which would frequently, though not always, obtain with certain kinds of sequential data.

### 4.1. Mathematical results

We are constructing what we hope to be a proof of uniqueness (R. A. Harshman and M. E. Lundy, unpublished manuscript) for solutions to the shifted factor model given in (4) or (7). The theorem can be summarized as follows. Consider any two alternative shifted factor representations of $\mathbf{X}$ (i.e. alternative factor profiles $\mathbf{A}$ vs $\overset{*}{\mathbf{A}}$, column weights $\mathbf{b}_j$ vs $\overset{*}{\mathbf{b}}_j$ and factor shift vectors $\mathbf{s}_j$ vs $\overset{*}{\mathbf{s}}_j$) that both produce the same array, so that

$$\mathbf{x}_j = \mathop{\mathcal{J}}_{\mathbf{s}_j} (\mathbf{A}) \mathbf{b}_j = \mathop{\mathcal{J}}_{\overset{*}{\mathbf{s}}_j} (\overset{*}{\mathbf{A}}) \overset{*}{\mathbf{b}}_j \tag{8}$$

In AIN this would be written as $x_{\mathrm{IJ}} = a_{[\mathrm{I}+\mathrm{s_{jr}}]\,\mathrm{R}} b_{\mathrm{JR}} = \overset{*}{a}_{[\mathrm{I}+\overset{*}{\mathrm{s}}_{\mathrm{jr}}]\,\mathrm{R}} \overset{*}{b}_{\mathrm{JR}}$.

If the structure of $\mathbf{X}$ fulfills certain conditions, the two representations can only differ in trivial ways. Specifically, they must be related as follows:

$$\overset{*}{\mathbf{B}} = \mathbf{B}\mathbf{P}\mathbf{\Delta}^{-1} \tag{9}$$

$$\overset{*}{\mathbf{A}} = \mathbf{A}\mathbf{P}\mathbf{\Delta} \tag{10}$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{\Delta}$ is diagonal (this assumes there is a convention setting the zero-shift locations).

*In other words, $x_{\mathrm{IJ}} = \left( \left( a_{[\mathrm{i}+\mathrm{s_{jr}}]\mathrm{r}} \right)_{\mathrm{IR}} b_{\mathrm{jR}} \right)_{\mathrm{J}}$.

#### 4.1.1. Requirements

The theorem's assumptions (i.e. some conditions sufficient to ensure this uniqueness) can be briefly summarized as follows: (i) **A** has full column rank; (ii) column shifting does not change the rank of **A**; (iii) the columns of **A** are not 'cyclic' (self-proportional when shifted); (iv) **X** includes a set of at least $R+1$ columns which has k-rank of $R$ and in which any two factors show independent shifts at least once.

The first three assumptions (maintenance of linear independence from the same and other factors after shifts, and non-cycling profiles) seem likely to be fulfilled in most real data situations. The maintenance of linear independence after shifting would usually be a natural consequence of the non-linearity of the transformations imposed by shifting. Non-linear factor transformations other than shifts (e.g. shape changes) might also fulfil these requirements, which raises potentially broad possibilities (see Section 6).

#### 4.1.2. Implications

If proven true, the theorem would support the idea that latent factor shifts can provide an added source of information that can be used to distinguish the factors. This would be analogous to the added information provided by the changes in factor weights across the third mode in multilinear models such as Parafac and Parafac2. Alternative 'rotation' and/or shifting of the factors (or combinations of shifting and rotation) could not reproduce the observed data.

#### 4.1.3. Necessary or sufficient conditions

The proof requires at least $R+1$ columns in **X** to obtain uniqueness of $R$ factors. We do not know whether this is necessary or whether fewer will do. (Recall, for example, that the first uniqueness proof for Parafac [42] needed to assume $R$ levels of every mode in order to prove uniqueness of $R$ factors, but the second proof [43] required only two levels of mode C for any number of factors*.) Independence of each factor's variation of shifts is also critical for uniqueness of the unconstrained version of the model. (Finally, it should be obvious that if any column of sequential factor loadings is 'cyclic' (i.e. proportional to a shifted version of itself), then the solution obtained for that factor is at most unique only up to the length of one such cycle. However, that may be all that is needed or meaningful in such conditions.)

### 4.2. Empirical results

We have performed some empirical tests of the uniqueness properties of the shifted factor model. Of course, computer experiments cannot prove uniqueness conclusively, but they can provide evidence in support of a conjecture (and of a proposed theorem). So far, our experimental question has been quite modest: does the two-way shifted factor model *ever* provide unique solutions?** Based on our empirical results, the answer seems quite likely to be 'yes'. Certainly, the particular cases that we have tested seem to be unique in the practical sense that no competing or alternative perfect-fitting solutions are observed under simulated data analytic conditions.

#### 4.2.1. Weighted factor case

Initial computational tests of uniqueness were carried out by Hong [44]. His methods and results have not been published elsewhere, but they contribute significantly to the evidence for uniqueness, so they are briefly summarized in Part II [40].

Additional experimental applications of the SFA model were carried out as part of our current study. The details are reported in Part II [40]. The new results further support the conjecture of identifiability of the SFA model. In addition, they show apparent uniqueness of the solution in conditions not covered by the mathematical analysis, in particular the 'pure shift' case.

#### 4.2.2 'Pure shift' two-way case

Having supported the hypothesis that incorporation of shift information into the factor model allows a unique two-way solution to be obtained, the question now becomes: what is the *strength* of the information provided by that portion of the data variance due to the factor shifts—to what degree might it determine the factors recovered by the two-way SF model, independently of information provided by variance resulting from factor size changes across levels of mode B? To explore this, we created a synthetic shifted factor data set using the same true values for the **A** and **S** parameters as in the $60 \times 75$ data for the weighted factor case (see Part II [40] for details), but using **B** weights that were all unity. Thus the variation across columns in the data is purely due to the independent shifting. Put another way, these data would have rank one without the shifts but have rank $J$ with the shifts. The latent SFA dimensionality was three.

Two versions of the shifted factor model were fit to these data, a restricted one and the more unconstrained one used above for the standard synthetic data. The restricted one was a 'pure shift' model. To fit it, mode B weights were held fixed at one, and **A** and **S** were fit as before. The result was that **A** and **S** were perfectly recovered. In the other test, where the fitting algorithm allowed the mode B weights to vary, we again observed perfect recovery of the factor loadings in **A**, **B** and **S** (after appropriate column scaling in **B** to set the weights to unity).

These results empirically demonstrate that the information in the data provided by independent shifting can be quite strong. Sometimes, shifting alone is sufficient to resolve the SFA factors and allow the model parameters to be uniquely determined. We currently have no mathematical analysis of the properties of SFA for 'pure shift' data.

## 5. THREE-WAY SHIFTED FACTOR MODELS

So far, the shifted factor principle has been discussed in terms of the shifted bilinear model (S-PCA/SFA). We now consider two basic multilinear (and quasi-multilinear) generalizations of this model to three-way data and higher-way data.

*A consideration of the capabilities of binocular vision suggests that fewer than $R+1$ levels might sometimes be enough to uniquely determine an SFA solution.

**Of course, the one-factor case is special. Data generated by only one factor have an essentially unique decomposition for all multilinear models of which we are aware, including the standard bilinear factor/component model. Nonetheless, although likely, it is not completely certain that SFA will be unique for one factor, since (a) it is only partly multilinear and (b) it has additional parameters and degrees of freedom compared with the bilinear model.

## 5.1. S-Parafac/S-Candecomp

Parafac/Candecomp is probably the simplest of the three-way (and multiway) models; this makes it useful to demonstrate the incorporation of shift capability into the higher-way domain. The acronyms stand for parallel factor analysis [21,42,45] and canonical decomposition [46] respectively. It is easy to specify the shifted factor generalization of the model using array notation, so we will do that first.

### 5.1.1. Using array notation

In array notation (AIN [38]) the Parafac/Candecomp model is written as

$$x_{\mathrm{IJK}} = a_{\mathrm{IR}} b_{\mathrm{JR}} c_{\mathrm{KR}} \tag{11}$$

To generalize (11) to incorporate shifting (at this point, of only one mode), we can either start with (4) and add factor loading parameters for the third mode, or start with (11) and add a shift parameter to the index I. One obtains

$$x_{\mathrm{IJK}} = a_{[\mathrm{I}+\mathrm{s}_{\mathrm{jr}}]\mathrm{R}}\, b_{\mathrm{JR}} c_{\mathrm{KR}} \tag{12}$$

### 5.1.2. Using matrix notation

There is no equally simple way to state, and then generalize, the Parafac model using matrix notation. This is understandable, since Parafac is three-way and matrix notation is inherently two-way. The problem is similar to that of representing the earth's surface on a flat sheet of paper. There have been two main methods that have been used to cope with this: (a) representing only a local region or (b) unfolding and flattening the curved surface. For a three-way array the analog of the first method is to work with a 'representative slice' [11,41]; the analog of the second is to use matrix 'unfolding' or matricization [41,47]. Generalization of the first method to shifted factor arrays is relatively straightforward; generalizing the second method is more difficult.

In the 'representative slice' formulation, one gives the latent structure of one slice, with a subscript appended to indicate repetition of similar slices across the third mode. Typically, the slice used is $\mathbf{X}_k$, a 'frontal slab' (see Reference [48], p. 7, and Reference [49], p. 231), and the model is written as

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{B}' \tag{13}$$

where $\mathbf{D}_k$ is a diagonal matrix containing the $k$th row of $\mathbf{C}$.

In our shifted factor generalization the diagonal weights will be selected from a *different* matrix, and so the greater transparency of angle bracket notation [41] will be helpful. (This notation simply consists of replacing the diagonalization operation usually represented as 'diag $(\cdot)$' with '$\langle\cdot\rangle$', i.e. an expression is enclosed in angle brackets to represent its diagonalization.) When (13) is rewritten in this way, it becomes

$$\mathbf{X}_k = \mathbf{A}\langle\mathbf{c}_k\rangle\mathbf{B}' \tag{14}$$

With shifted factor analysis the representative slice cannot be a frontal slice because of the convention that **A** is the sequential 'shifted' mode and **B** is the 'shifting' mode; thus shifts in **A** change from one column of **B** to the next. This makes it impossible to use a fixed **A** for all columns of **B** simultaneously, as was done in (13) and (14).

If, however, we consider the subset of all $x_{ijk}$ values that *do* share a given shifted version of **A**, we obtain the $I \times K$ matrix at level $j$ of mode B. This is a 'lateral slab' or lateral slice of $\underline{\mathbf{X}}$ (see Reference [48], p. 7, and Reference [49], p. 231). As a basis for shift generalization we rewrite the standard unshifted Parafac model as

$$\mathbf{X}_j = \mathbf{A}\langle\mathbf{b}_j\rangle\mathbf{C}' \tag{15}$$

where **A** and **C** are factor loading matrices for modes A and C respectively and $\langle\mathbf{b}_j\rangle$ is a diagonal matrix containing the $j$th row of **B**.

Now we can write a matrix version of the generalization to shifted Parafac1. It is

$$\mathbf{X}_j = \mathcal{J}_{\mathbf{s}_j}(\mathbf{A})\langle\mathbf{b}_j\rangle\mathbf{C}' \tag{16}$$

(In the following discussion we will sometimes refer to the model given in (16) or (12) as S-Parafac or, more precisely, S-Parafac1 and compare it with e.g. S-T3, S-Paratuck2, S-DEDICOM, etc.)

It is more difficult to generalize the three-way factor model to shifted form when it is written using the unfolding/matricizing approach. In the unshifted version the data array is converted into a matrix by adjoining successive slices to produce one long two-way data set (see e.g. Reference [47]). Then the Parafac model can be expressed in matrix terms either by means of the Khatri–Rao* columnwise Kronecker product '$\odot$' [11] or by means of the standard Kronecker product '$\otimes$' combined with an enhanced notation** (see Reference [47], p. 109). The first method gives

$$\mathbf{X}^{(\mathrm{I}\times\mathrm{JK})} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})' \tag{17}$$

and the second

$$\mathbf{X}_{\mathrm{a}} = \mathbf{A}\mathbf{I}_{\mathrm{a}}(\mathbf{C}' \otimes \mathbf{B}') \tag{18}$$

It would be tempting to jump to the conclusion that the corresponding shifted factor generalizations are simply $\mathbf{X}^{(\mathrm{I}\times\mathrm{JK})} = \mathcal{J}_{\mathbf{s}_j}(\mathbf{A})(\mathbf{C} \odot \mathbf{B})'$ and $\mathbf{X}_{\mathrm{a}} = \mathcal{J}_{\mathbf{s}_j}(\mathbf{A})\mathbf{I}_{\mathrm{a}}(\mathbf{C}' \otimes \mathbf{B}')$, but there are two difficulties. First, there is no longer a single $j$ for each shifted version of **A**. This is relatively easy to overcome, by simply defining a new column subscript that combines both $j$ and $k$ and then establishing appropriate values in the associated row vectors of an extended **S**. The second problem is more difficult. After unfolding the array, we have a two-mode model that has the same structure as (1), and so we encounter the same difficulty that arose when we tried to generalize (1): there is no fixed single **A** in the shifted version. By defining new matrix operations, it may be possible to deal with this, but this would take us beyond the scope of this paper.

## 5.2. Tucker models

The other fundamental three-way model family in current research is Tucker's three-mode factor analysis family, which includes T3, T2 and related models [48] (including, more recently, constrained Tucker; see e.g. Reference [50]). Many of the issues raised above for Parafac apply here as well and so need not be repeated.

*Khatri-Rao-Bro

**The subscript 'a' identifies the unmodified mode.

### 5.2.1. S-T3 and S-T2

By arranging one's data so that mode A is the sequential mode and mode B is the 'shift controlling' mode, it is straightforward to obtain a simple version of the shifted Tucker T3 (S-T3) model. In array notation the model might be written as

$$x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jP}}]\mathrm{P}} b_{\mathrm{JQ}} c_{\mathrm{KR}} g_{\mathrm{PQR}} \tag{19}$$

When written in representative slice form, using 'encapsulated summation' [41], and switching B and C as in (16), this is

$$\mathbf{X}_j = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\left(\sum_q b_{jq}\mathbf{G}_q\right)\mathbf{C}' \tag{20}$$

Similarly, S-T2 becomes

$$x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jP}}]\,\mathrm{P}}\, c_{\mathrm{KR}}\, g_{\mathrm{PJR}} \tag{21}$$

in AIN and

$$\mathbf{X}_j = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\mathbf{G}_j\mathbf{C}' \tag{22}$$

in representative slice form. The above representations have only one shifted mode, but there are, no doubt, other interesting possibilities that we have not explored.

### 5.2.2. Uniqueness

It is interesting to note that while the unshifted versions of these Tucker models are under-identified (unless special restrictions are imposed as in Reference [50]), it seems quite likely that their shifted versions will have identifiable parameters. This is because these models can be written such that each slice is a shifted two-way model, and we have strong evidence (reviewed above) that such two-way models have a unique solution. If so, S-T3 and S-T2 would combine the Tucker models' greater structural generality (relative to Parafac/Candecomp) with the desirable property of identifiable parameters, which previously was possessed by only the more restricted models.

## 5.3. Other possibilities for future exploration

Table I shows standard and generalized forms of the direct fit models discussed so far. Of course, this table does not exhaust the possibilities; from these examples, however, one can easily see how other multilinear and quasi- or semi-multilinear models could be generalized to allow shifting of latent variables. Now we will briefly venture beyond the more familiar models discussed in the table and mention some other possible shifted factor models that have not yet been explored.

**Table I.** Direct fit models: unshifted and shifted representations

| Model | Matrix notation | Array index notation |
|---|---|---|
| PCA | $\mathbf{X} = \mathbf{A}\mathbf{B}'$ | $x_{\mathrm{IJ}} = a_{\mathrm{IR}} b_{\mathrm{JR}}$ |
| | $\mathbf{X} = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\mathbf{B}'$ | $x_{\mathrm{IJ}} = a_{[\mathrm{I}+s_{\mathrm{jr}}]\mathrm{R}} b_{\mathrm{JR}}$ |
| Parafac1/ | $\mathbf{X}_j = \mathbf{A}\langle\mathbf{b}_j\rangle\mathbf{C}'$ | $x_{\mathrm{IJK}} = a_{\mathrm{IR}} b_{\mathrm{JR}} c_{\mathrm{KR}}$ |
| Candecomp | $\mathbf{X}_j = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\langle\mathbf{b}_j\rangle\mathbf{C}'$ | $x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jr}}]\mathrm{R}} b_{\mathrm{JR}} c_{\mathrm{KR}}$ |
| T3 | $\mathbf{X}_j = \mathbf{A}\left(\sum_q b_{jq}\mathbf{G}_q\right)\mathbf{C}'$ | $x_{\mathrm{IJK}} = a_{\mathrm{IP}} b_{\mathrm{JQ}} c_{\mathrm{KR}} g_{\mathrm{PQR}}$ |
| | $\mathbf{X}_j = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\left(\sum_q b_{jq}\mathbf{G}_q\right)\mathbf{C}'$ | $x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jp}}]\mathrm{P}} b_{\mathrm{JQ}} c_{\mathrm{KR}} g_{\mathrm{PQR}}$ |
| T2 | $\mathbf{X}_j = \mathbf{A}\mathbf{G}_j\mathbf{C}'$ | $x_{\mathrm{IJK}} = a_{\mathrm{IP}} c_{\mathrm{KR}} g_{\mathrm{PJR}}$ |
| | $\mathbf{X}_j = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\mathbf{G}_j\mathbf{C}'$ | $x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jp}}]\mathrm{P}} c_{\mathrm{KR}} g_{\mathrm{PJR}}$ |

### 5.3.1. Models for cross-product or covariance data

The models proposed so far have all been what Kruskal [51] calls 'direct fit' models; that is, they are applied directly to the data array. In contrast to these are the 'indirect fit' models—models for the structure underlying sets of quantities derived from the data, such as covariances or other cross-product arrays. Included in these are standard factor analysis of correlation or covariance matrices, both inner-product and outer-product versions, and three-way models such as Parafac2 and Tucker's three-mode multidimensional scaling (TMMDS). Some preliminary exploration indicates that there are a number of interesting things that result from applying the shifted factor principle to these cross-product-type matrices, but it is too large a subject to take up here.

Two less obvious directions of possible further development that we would like to mention here are (a) models with two sequential modes and in which one wants to make allowance for shifts in both these modes, and (b) models involving shifts in a plane or space rather than along a line. Currently, neither of these directions has been explored beyond the formulation of the basic idea.

### 5.3.2. Doubly shifted factor structures

Some types of data involve more than one sequential mode. Consider, for example, a sound spectrogram. It gives energy values at each of many successive frequencies for each of many successive times. Now, if, in addition, an investigator simultaneously records data from these sound sources using microphones placed at several different locations, we obtain a three-mode data set that could be organized as frequency × time × location.

Suppose these data have structure that can be described in terms of latent variables. For example, suppose there are several sound sources (e.g. several airplanes) and that we want to separate out and measure their individual characteristics based only on the mixtures of sound arriving at the different microphones. We can consider these sound sources to be latent causal factors producing the spectral mixtures observed at each microphone. Each source/factor would have one sequential loading profile in the frequency mode and one in the time mode. Either or both of these profiles could be shifted when the latent structure at one recording location is compared with that at another, and the pattern of variations (across recording locations) in one factor's shift sizes could be independent of the patterns for other factors. Across the frequency mode the spectral profile corresponding to a given airplane would shift position as a function of the velocity of the source in the direction of the receiver (Doppler shift). Across the time mode the observed temporal position of changes in the intensity of a given factor could be shifted from one recording location to the next as a function of the distance between the source and the receiver's location (owing to signal propagation lag).

We can imagine what some models for situations such as this might look like. For example, if the sources are distant

enough from the microphones*, one might consider a shifted Parafac model such as

$$x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jr}}]\mathrm{R}} b_{\mathrm{JR}} c_{[\mathrm{K}+\widehat{\mathbf{s}}_{\mathrm{jr}}]\mathrm{R}} \tag{23}$$

where mode A is frequency, mode B is location, mode C is time and the shift values $s_{jr}$ and $\widehat{\mathrm{s}}_{\mathrm{jr}}$ are taken from two matrices of shift values, $\mathbf{S}$ for mode A and $\widehat{\mathbf{S}}$ for mode C. This model can be written using the shift operator by considering a representative lateral slice:

$$\mathbf{X}_j = \mathcal{S}_{\mathbf{s}_j}(\mathbf{A})\langle \mathbf{b}_j \rangle (\mathcal{S}_{\hat{\mathbf{s}}_j}(\mathbf{C}))' \tag{24}$$

In similar fashion a doubly shifted version of the S-T3 model might be written in AIN as

$$x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{\mathrm{jp}}]\mathrm{P}} b_{\mathrm{JQ}} c_{[\mathrm{K}+\widehat{\mathrm{s}}_{\mathrm{jr}}]\mathrm{R}} g_{\mathrm{PQR}} \tag{25}$$

We find this model difficult to write using standard matrix notation plus the shift operator.

For those less familiar with array notation, we can write an expression for a 'typical element' in the doubly shifted S-T3 as

$$x_{ijk} = \sum_p \sum_q \sum_r a_{(i+s_{jp})p}\, b_{jq}\, c_{(\mathrm{k}+\widehat{\mathrm{s}}_{\mathrm{jr}})r}\, g_{pqr} \tag{26}$$

As usual, array and scalar notation have a very similar structure, but the array version represents the entire array, and can be manipulated as such [38].

These double-shift models seem interesting from a formal point of view, and potentially useful, but this is another direction that we have not actually explored.

### 5.3.3. *Spatial shifts*

Time series and spectra are one-dimensional sequences. Other types of data are sequential in a higher-dimensional continuum, such as in a plane (e.g. image) or three-space (e.g. brain or body 'scan', seismic array data, atmospheric pollution patterns). An analysis of images into latent image components might want to allow for modest spatial shifting of the latent images along any direction in the image plane. More generally, it might be useful to have models that could easily deal with factors that shift within a multi-dimensional space. One possible approach is to change a loading *matrix* such as $a_{\mathrm{IR}}$ into a loading *array* $a_{\mathrm{IKR}}$. In the case of two-dimensional sequences, each mode A loading for a given factor would be identified by its joint position on two co-ordinates. It seems quite difficult to state such a model in matrix terms, but in array notation it would have the form

$$x_{\mathrm{IJK}} = a_{[\mathrm{I}+s_{1\mathrm{jr}}][\mathrm{K}+s_{2\mathrm{jr}}]\mathrm{R}} b_{\mathrm{JR}} \tag{27}$$

Here the size and direction of factor shifts on the plane would be given by the vector $[s_{1\mathrm{jr}}\ s_{2\mathrm{jr}}]$. The shifts for each mode A factor would be represented by two columns.

For models such as this, there would be a range of valid alternative linear combinations within each factor's co-ordinate pair, corresponding to 'rotations' of co-ordinate axes in the image plane. This harmless pairwise indeterminacy should not affect the unique isolation of meaningful components defined by the between-pair relationships among the factors. And the added information provided by the shifting might allow the columns of $b_{\mathrm{JR}}$ to be uniquely determined.

*Different models would seem to be necessary when the sound sources are relatively close to the microphones.

## 6. DISCUSSION

It is clear that the basic two-way model for factor/component analysis can be extended to accommodate shifts in latent factors, as can the basic three-way models Parafac/Candecomp and Tucker T3 or T2. Our work so far indicates that this generalized model provides (essentially) unique solutions, given certain conditions such as independent factor shifts. Apparently, the factor shifts can add enough information distinguishing one factor from another to allow the factors to be uniquely recovered, much as changes in factor variance across the third mode allow the Parafac/Candecomp model to determine a unique, best-fitting set of factors. The difference, of course, is that the shift parameters fit the non-linearity (or non-multilinearity) in the data that is due to positional shifts, while the Parafac third-mode weights describe linear variations. Another example, although not fully comparable, of how the incorporation of information on non-linear factor changes can provide unique resolution of factors in two-way data, is provided by the incorporation (via constraints) of knowledge that a particular mode will show exponential decay of loading magnitudes (see References [52–57] and references cited therein). In the case of three-way shifted factor models, both the shift structure and the three-way multilinear structure can contribute to uniqueness. This could strengthen the uniqueness properties of the solution in cases where independent variation of the factors in either of these two sources of added information was weak or not present for some pairs of factors.

### 6.1. Shape changes

With some kinds of data, factor position shifts are often associated with shape changes. For such data the simple linear version of the models described here provides only part of the needed generalization.

How can this further complication be dealt with? Tauler [24] suggests that we resort to flexible curve resolution methods. Although these methods do not—unaided—provide the uniqueness of multilinear models, he points out that we can often strengthen them by applying other principles such as selectivity, positivity constraints, etc. to recover the correct solution. Wulfert *et al*. [28,58], on the other hand, focus on the use of prior knowledge about the nature of the shape changes. A (perhaps modified) version of their polynomial curve fit and banded matrix approach might provide one possible mechanism by which shape changes could be accommodated within the present context.

It is possible that, when only a few parameters are required in order to adequately accommodate the shape variations likely to occur in a particular situation, then their estimation could be included as part of the data-fitting process. The main differences between this approach and that of Wulfert *et al*. [28,58] might only be that (a) the non-linear shape changes

would be characterized in a somewhat more general and abstract form, and (b) the parameter values along the shape change continuum would not be known in advance but instead would be estimated as part of the curve-fitting process. However, in a more general 'black box' approach the shape change parameters introduced into the model might control fundamental geometric characteristics such as skewness or kurtosis of the 'humps' in a given factor.

The Appendix includes a definition of shifting in terms of a general functional relationship between an index on the unshifted factor and the corresponding index on the shifted one, as in

$$y_n = x_{f(m,p)}$$

that is, where

$$n = f(m, p)$$

and $m$ can vary in either discrete or continuous fashion. In the cases considered earlier in this paper, we have used the simple relationship

$$n = m + p$$

Two straightforward examples of possible kinds of generalization would be

$$f(m, p) = m + h(p) = m + ap + bp^2 + \cdots$$

or

$$\begin{aligned} f(m, p) &= m + h(p_1, p_2, \ldots) \\ &= m + ap_1 + bp_2p_3 + \cdots \end{aligned}$$

Depending on the functional relationship, it may or may not be possible to approximate variation that includes both shifting and shape changes.

The addition of parameters to describe shape change might add too many degrees of freedom to the SF model and so interfere with its uniqueness, but not necessarily. It seems plausible that the uniqueness properties of the original SF model could be retained so long as the specified factor transformations have the same effects on linear independence as those induced by 'simple' position shifts, at least to the extent that assumptions (i)–(iii) of the theorem described earlier are still applicable. If these assumptions still do apply, then the mathematical uniqueness and other properties explored by the proofs could be carried over—in whole, or at least in part—to the new domain that includes shape change. Of course, these extended models do not provide any improved identifiability if the data do not have the additional kind of factor variation specified in the model. Also, one must be careful to avoid indeterminacies *within* the non-linear part of the model because of possible trade-offs among the various added parameters.

## 6.2. Parametrized non-linear changes

At a higher level of generality, one might consider the shifted factor model presented here as a specific case of a broader principle: one can strengthen a model and sometimes obtain uniqueness of solutions by adding parameters to the model to fit additional, previously ignored, characteristics of factor variation. While here we have added parameters specifically to fit position shifts (and possibly certain shape shifts) of the factors, it may be that *any* non-linear factor changes *that can be described by a few parameters* could create newly identifiable models, provided that the factor changes incorporated into the model fulfil certain conditions. If this conjecture is correct, then it provides a general approach to finding and formulating identifiable models; perhaps one could call it 'Paramfac' (for parametric factor analysis).

Put another way, the operator symbol $\mathcal{S}(\cdot)$ need not refer to *shifts* of factors or even be limited to sequentially organized data; it can be considered abstractly as representing an operation that changes factor loading vectors. A previously under-identified model extended to incorporate such transformations would not only have the virtue of fitting a new kind of variation, but also it could become identifiable—have uniquely determined solutions—whenever the nature of the added factor changes specified by $\mathcal{S}(\cdot)$ fulfil certain algebraic conditions (e.g. maintenance of certain linear independence and k-rank properties of sets of factors after applications of the transformation, as assumed in the theorem mentioned earlier). Models that were previously under-identified, such as the bilinear factor model and the fully general form of Tucker's T3 model, might in this way gain desirable uniqueness properties.

## Acknowledgements

The first author was supported by Natural Sciences and Engineering Research Council of Canada research grant OGP-000-7896. We thank Age Smilde, Rasmus Bro and others for their encouragement.

# APPENDIX. ARRAY SHIFTING: DEFINITIONS, ASSUMPTIONS AND NOTATION

## A.1. The shift operation

We start with an abstract definition of shifting that allows the sequential mode to consist of either vectors with discrete elements or continuous functions, but then continue the discussion concentrating on the discrete case, where the shifts are integers and refer to subscript position changes.

### *A.1.1. Shifting (continuous or discrete)*

Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors. The vector $\mathbf{y}$ is a *shifted* version of $\mathbf{x}$ if, for all $m$,

$$y_n = x_m, \quad \text{where} \quad n = f(m, s) \tag{28}$$

Here $s$ is a parameter (or parameter set) controlling the amount (and/or nature) of the shift.

This 'shifting' relationship can also be written in terms of the vectors as

$$\mathbf{y} = g(\mathbf{x}, s) \tag{29}$$

The function $g$ *shifts* $\mathbf{x}$ to produce $\boldsymbol{y}$ by transforming its index values in a manner determined by $s$.

Generalization to arrays of different orders is straightforward; it involves shifts of subarrays, controlled by parameters that may have more than one index. This will be explained in more detail below.

### *A. 1.2. Continuous case*

The function $f$ is monotonic and usually linear; typically, $f(m,s) = m + s$. Other relationships are possible, however;

for example, $f(m,s) = m + s_1 + s_2 e^{1/(m-s)} \ldots$ . As shown, the parameter $s$ can be a set of values.

If $m$ is a point on a section of the real line over some interval $m_{\min} < m < m_{\max}$, then the index values that $n$ takes on are the result of evaluating the function $f$ over that interval, and this result may or may not be continuous. Sequential manifest vectors that consist of smooth functions can be decomposed into combinations of latent factors that are themselves smooth functions. On the other hand, vectors in which $m$ and $n$ take on integer values are also covered by the above definition.

### *A.1.3. Discrete case*

Suppose we have a vector $\mathbf{x}$ that contains $I$ elements $x_i$. One way that we can express the contents of $\mathbf{x}$ is

$$\mathbf{x} = [x_1, x_2, \ldots, x_i, \ldots, x_I] \tag{30}$$

Our notion of discrete 'shifting' relies on this type of spatial representation, in which the values are thought of as assigned to successive locations along a left–right or up–down sequence. The phrase 'shifting $\mathbf{x}$' is (in the discrete case) short for 'shifting the elements in vector $\mathbf{x}$ along the subscript sequence', which could be thought of as moving each of the values in the vector so many places to the right or to the left in the sequence. More abstractly, it is equivalent to systematically increasing or decreasing the quantity assigned to the subscript $i$ for each value $x_i$ in the vector.

The result of the shift operation is a new set of sequential values that could be used in some expression or assigned to a new variable. For example, if $\mathbf{y}$ is the result of shifting $\mathbf{x}$ by $s$ places, then each element of $\mathbf{y}$ can be expressed as

$$y_i = x_{i+s} \tag{31}$$

A.1.3.1 Ends of vectors. Several decisions must be made and conventions established to remove ambiguities in the notion of 'shifting'. Most of these have to do with what happens at the ends of the shifted vector. We consider three basic approaches: (a) for the purposes of the discussion, the vectors can be considered unbounded in both directions ('infinite window'); (b) the vectors resulting from shifts can be considered to consist of those elements that are 'within the window' both before and after the shift ('shrinking window'); and (c) the vectors can be considered to have fixed length, but with contents determined by the position of a 'window' along a longer version of that vector ('sliding window'). The choice among these interpretations will often not matter, but it can be important in certain circumstances, for example when considering the preservation of vector equality after shifting (see below).

We have chosen to concentrate here on (c), the fixed length or 'sliding window' interpretation.* This leads us to define a shift operation that has the following properties. The shift operation does not change the length of the vector to which it is applied. For example, if the length of $\mathbf{x}$ is $I$, the length of the shifted version $\mathbf{y}$ is also $I$. After a shift of size $s$, the relationship between the elements of $\mathbf{x}$ and $\mathbf{y}$ is as follows. If $s = 0$, then $\mathbf{y} = \mathbf{x}$. If $s > 0$, then

$$\begin{aligned}\mathbf{y} &= [y_1, y_2, \ldots, y_I] \\ &= [x_{1+s}, x_{2+s}, \ldots, x_I, new_1, new_2, \ldots, new_s]\end{aligned} \tag{32}$$

If $s < 0$, then

$$\begin{aligned}\mathbf{y} &= [y_1, y_2, \ldots, y_I] \\ &= [new_s, \ldots, new_2, new_1, x_1, x_2, \ldots, x_{I-abs(s)}]\end{aligned} \tag{33}$$

To specify where these new values will come from, any vector to be shifted is, for the purpose of such shifting, thought of as a part of a longer vector. For vector $\mathbf{x}$, we define a vector '$\mathbf{x}$-long' of length $L$, where $L > I$, and write it as $\underset{\leftrightarrow}{\mathbf{x}}$. Thus we have

$$\underset{\leftrightarrow}{\mathbf{x}} = \left[\underset{\leftrightarrow}{x}_1, \underset{\leftrightarrow}{x}_2, \ldots, \underset{\leftrightarrow}{x}_I, \ldots, \underset{\leftrightarrow}{x}_L\right] \tag{34}$$

(We name the 'long' vectors using a modified version of the shorter vector's name, instead of a new symbol such as u, in order to emphasize the close relation between the longer and shorter versions of the vector and to provide a convention for generating an immediately recognizable name for the longer version of any vector.)

We next identify the position of $\mathbf{x}$ in $\underset{\leftrightarrow}{\mathbf{x}}$. We define the scalar $p$ (for $\mathbf{x}$-position) such that

$$\underset{\leftrightarrow}{x}_p = x_1 \tag{35}$$

and so

$$[x_1, x_2, \ldots, x_i, \ldots, x_I] = \left[\underset{\leftrightarrow}{x}_p, \underset{\leftrightarrow}{x}_{p+1}, \ldots, \underset{\leftrightarrow}{x}_{p+i-1}, \ldots, \underset{\leftrightarrow}{x}_{p+I-1}\right] \tag{36}$$

This provides a source for the $new_1$, $new_2$, etc. in (32) and (33). Now we have

$$\begin{aligned}\mathbf{y} &= [x_{1+s}, x_{2+s}, x_{3+s}, \ldots, x_I, new_1, new_2, \ldots, new_s] \\ &\equiv \left[\underset{\leftrightarrow}{x}_{p+s}, \underset{\leftrightarrow}{x}_{p+1+s}, \underset{\leftrightarrow}{x}_{p+2+s}, \ldots, \underset{\leftrightarrow}{x}_{p+(I-1)+s}\right]\end{aligned} \tag{37}$$

Elements beyond the ends of the two 'windows' (i.e. other elements in $\mathbf{y}$) are not considered.

Anomalies would arise from a vector $\mathbf{v}$ (or, equivalently, the window defining the vector) being shifted beyond the end of its corresponding long version $\underset{\leftrightarrow}{\mathbf{v}}$. This is excluded in one of two ways: (1) for mathematical purposes, all 'long' vectors ($\underset{\leftrightarrow}{\mathbf{x}}$, $\underset{\leftrightarrow}{\mathbf{y}}$, etc.) are considered infinite in both directions (of course, this requires that zero and negative subscripts are allowed); or (2) the long vectors $\underset{\leftrightarrow}{\mathbf{x}}$, $\underset{\leftrightarrow}{\mathbf{y}}$, etc. are considered finite but, by assumption, are *long enough* relative to their corresponding short vectors to prevent the window that defines the short vector from going beyond either end of the long vector in any shift or cumulative sum of shifts under consideration in a given context (e.g. in a given proof, analysis, etc.). In other words, the long versions of shifted vectors are by definition long enough to ensure that there always exists an element $\underset{\leftrightarrow}{x}_{p+(I-1)+s}$ for any occurring positive shift or positive sum of shifts $s$, and likewise an element $\underset{\leftrightarrow}{x}_{p-s}$ for any occurring negative shift or sum of shifts $-abs(s)$.

Although the results in the preceding subsections have been stated in terms of single vectors, the same idea

*It is easier to define and consider the properties of vector ends in the discrete case, so we will discuss this case here. The generalization to the continuous case is handled only by implication.

generalizes in a natural way to shift operations on matrices, arrays, multiple columns of a matrix, etc. The definitions below are constructed in more general form to include shifting of various subarrays inside larger arrays, including shifting of vector segments.

## A.2. Notation for expressing the shift operation

To incorporate the shift operation into models, algorithms and proofs, we need a precise way of specifying shifts. Two approaches will be discussed here: (i) for array index notation (AIN) [38], nothing new is needed—we simply review and perhaps clarify the interpretation to be given to certain subscript expressions; (ii) for standard matrix notation we define a 'shift operator' and set out rules for its use.

### *A.2.1. Expressing the shift operation in array index notation*

If one is using AIN [38], there is no need to define new symbols to represent shifted objects. AIN explicitly incorporates index information into the array names, and so shifted objects can be represented by simply indicating a function altering the array index information of the originally unshifted object. For example, if the vector $y_{\mathrm{I}}$ is the result of shifting vector $x_{\mathrm{I}}$ by $s$, we can simply write

$$y_{\mathrm{I}} = x_{[\mathrm{I}+\mathrm{s}]} \tag{38}$$

For simplicity, let us return to a discrete case, where I consists of integer subscript values. To represent the column-shifted matrix $\mathscr{S}_{\mathrm{s}_j}(\mathbf{X})$, the matrix equation

$$\mathbf{Y} = \underset{\mathrm{s}_j}{\mathscr{S}}(\mathbf{X})$$

becomes the AIN equation

$$y_{\mathrm{IJ}} = x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{J}} \tag{39}$$

When interpreting (39), we make use of the AIN principle that multiple occurrences of the same non-italic index symbol within a given AIN expression are interpreted as multiple occurrences of the *same* generic index value (Rule 2a of Reference [38]). Here the match is between the s subscript j and an implied element of J. No new principle is really needed, since this type of compact expression can always be replaced by a longer one that uses parentheses in the same way as in composite elements (Rule 3b of Reference [38]). For example, the right-hand side of (39) may be defined as

$$x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{J}} \equiv \left(x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{j}}\right)_{\mathrm{J}} \tag{40}$$

It follows that, for each column, $y_{\mathrm{Ij}} = x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{j}}$, and so

$$\begin{aligned} y_{\mathrm{IJ}} &= x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{J}} \\ &= \left[x_{[\mathrm{I}+\mathrm{s}_1]1}\ x_{[\mathrm{I}+\mathrm{s}_2]2}\ x_{[\mathrm{I}+\mathrm{s}_3]3}\ \cdots\ x_{[\mathrm{I}+\mathrm{s}_\mathrm{J}]}\right] \end{aligned} \tag{41}$$

(recall that J is the upper end of the range of j). In other words, as J goes through its range of values, the j on the shift size $\mathrm{s}_\mathrm{j}$ goes through the same values.

A.2.1.1. Examples. To represent a three-way array in which lateral $(I \times K)$ slabs are shifted along the levels of mode A, we could write

$$y_{\mathrm{IJK}} = x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{JK}} \tag{42}$$

On the other hand, for one with individually shifted column fibers, we could write

$$y_{\mathrm{IJK}} = x_{[\mathrm{I}+\mathrm{s}_\mathrm{jk}]\mathrm{JK}} \tag{43}$$

If we wanted to very explicitly express the source of the values of the shift subscripts, as in (40), Equations (42) and (43) could be written as $y_{\mathrm{IJK}} = \left(x_{[\mathrm{I}+\mathrm{s}_\mathrm{j}]\mathrm{jK}}\right)_{\mathrm{J}}$ and $y_{\mathrm{IJK}} = \left(x_{[\mathrm{I}+\mathrm{s}_\mathrm{jk}]\mathrm{jk}}\right)_{\mathrm{JK}}$ respectively.

To demonstrate the flexibility of this kind of shift notation, we consider the case where the elements in these $(I \times K)$ slabs are shifted *diagonally* within the plane in which the slabs lie. These shifts could be written as

$$y_{\mathrm{IJK}} = \left(x_{[\mathrm{I}+\mathrm{s}_{1\mathrm{j}}]\mathrm{j}[\mathrm{K}+\mathrm{s}_{2\mathrm{j}}]}\right)_{\mathrm{J}} \tag{44}$$

Here the shift values are stored in a $J \times 2$ matrix with the first column giving the shift amounts in mode A and the second column giving the shift amounts in mode C. Once again using the simplified notation as in (40) Equation (44) can be expressed as

$$y_{\mathrm{IJK}} = x_{[\mathrm{I}+\mathrm{s}_{\mathrm{j}1}]\mathrm{J}[\mathrm{K}+\mathrm{s}_{\mathrm{j}2}]} \tag{45}$$

A.2.1.2. Summation of matching indices. A final note: when performing summation over matching indices (as is done in AIN following the 'Einstein convention' [38]), the matching is done on the index value *before* it is shifted or otherwise transformed. Thus

$$\begin{aligned} u_{[\mathrm{P}+7]}v_{[\mathrm{P}+9]} &= \left(u_{[\mathrm{p}+7]}\right)_{\mathrm{P}}\ \left(v_{[\mathrm{p}+9]}\right)_{\mathrm{P}} \\ &= \sum_{p} u_{[p+7]}v_{[p+9]} = w \end{aligned} \tag{46}$$

This point is only going to matter, or be comprehensible, to those readers who make use of array notation when working with shifted vectors or arrays.

### *A.2.2. Expressing the shift operation in matrix notation: the shift operator*

A.2.2.1. Definition of the operator. In matrix notation the shifting (subscript incrementing or decrementing) operation will be designated by the symbol

$$\mathscr{S} \tag{47}$$

which is a script 'S' (standing for 'shift'). We define a shift $s$ for a single element in $\mathbf{x}$ as

$$\underset{\mathrm{s}}{\mathscr{S}}(x_i) \equiv x_{i+s}$$

and a shift $s$ for the vector $\mathbf{x}$ as

$$\underset{s}{\mathscr{S}}(\mathbf{x}) \equiv x_i \rightarrow x_{i+s} \quad \text{for} \quad i = 1,\ 2,\ \ldots,\ I \tag{48}$$

More generally, we now define a shift operator (or shifting function) $\mathscr{S}(\cdot)$ that takes as arguments an array object to be shifted, a (smaller) object containing shift sizes, and an expression defining the shifted part.

The notation has the general form

$$\underset{\beta}{\overset{\gamma}{\mathcal{S}}}(\alpha) \tag{49}$$

where $\alpha$ is the array to be operated on, $\beta$ contains the shift sizes and $\gamma$ defines the 'unit of shifting' (the size and shape of a partition in $\alpha$ containing elements that are shifted together as one unit). The arguments may in general be arrays of any order (scalar, vector, matrix or higher-way), except that the sum of the orders of $\beta$ and $\gamma$ should equal the order of $\alpha$.

The fully explicit form of notation defined in (49) is not always needed. When the information that would be provided by $\gamma$ and/or $\beta$ is already clear from the context, these arguments could be omitted. Sometimes $\gamma$ need not be specified if $\alpha$ and $\beta$ are: its shape (order) can be deduced from their order numbers, and its size from $\beta$ or the context. Likewise, $\beta$ can be deduced from $\alpha$ and $\gamma$. However, when the order of $\alpha$ is greater than two, explicitly representing $\beta$ and $\gamma$ is usually recommended.

Sometimes a further simplification is used: $\beta$ is expressed in 'shorthand' form, where only the appropriate subscript is given, e.g. $\mathcal{S}_j(\cdot)$. In this case the shift values are assumed to reside in part of an array of shift values that should be obvious by the context (often a vector taken from the shift matrix **S**). The part of the subarray from which the values are taken is designated by the subscript(s) used in the 'shorthand' (e.g. often $\mathcal{S}_j \equiv \mathcal{S}_{\mathbf{s}_j}$).

A.2.2.2. Examples. A few examples will help clarify the meaning and usage of the shift operator.

A.2.2.2.1. Elements and segments. As noted earlier, a shift applied to a single element in a vector **x** may be specifically denoted as $\mathcal{S}_5(x_2) = x_7$, for example, or more generally as $\mathcal{S}_{w+y}(x_i) = x_{i+w+y}$.

A.2.2.2.2. Vectors. The simplest generalization consists of shifting the position of elements in a single vector. To represent the vectors that result when each element in **x** is shifted by, say, five places or by $u + v + 1$ places, we have

$$\underset{5}{\mathcal{S}}(\mathbf{x}) \equiv \begin{bmatrix} \vdots \\ x_{1+5} \\ x_{2+5} \\ \vdots \\ x_{i+5} \\ \vdots \end{bmatrix} \quad \text{and} \quad \underset{u+v+1}{\mathcal{S}}(\mathbf{x}) \equiv \begin{bmatrix} \vdots \\ x_{1+u+v+1} \\ x_{2+u+v+1} \\ \vdots \\ x_{i+u+v+1} \\ \vdots \end{bmatrix} \tag{50}$$

Note that $\gamma$ has been omitted here as unnecessary; in its fully explicit form the expression would be $\mathcal{S}^{\mathbf{x}}_{u+v+1}(\mathbf{x})$ to specify that **x** is shifted.

A.2.2.2.3. Matrices. A column-shifted *matrix* can be represented by assembling vectors that are expressed as in (50) into successive column positions in a partitioned matrix. However, it is more flexible and compact to use the potential for greater generality implicit in (49) to express the application of successive elements from a vector of shifts to the successive columns of the matrix. Let **X** be an $I \times J$ matrix and **v** be a vector of length $J$, with each element of **v** giving the shift size for the corresponding column in **X**. Then we have three alternative representations:

$$\begin{aligned} \underset{\mathbf{v}}{\mathcal{S}}(\mathbf{X}) &\equiv \left[ \underset{v_1}{\mathcal{S}}(\mathbf{x}_1) \,|\, \underset{v_2}{\mathcal{S}}(\mathbf{x}_2) \,|\, \cdots \,|\, \underset{v_j}{\mathcal{S}}(\mathbf{x}_j) | \cdots \right] \\ &\equiv \begin{bmatrix} \vdots & \vdots & & \vdots & \\ x_{1+v_1} & x_{1+v_2} & & x_{1+v_j} & \\ x_{2+v_1} & x_{2+v_2} & & x_{2+v_j} & \\ \vdots & \vdots & \ddots & \vdots & \\ x_{i+v_1} & x_{i+v_2} & & x_{i+v_j} & \\ \vdots & \vdots & & \vdots & \ddots \end{bmatrix} \end{aligned} \tag{51}$$

Again, $\gamma$ is unnecessary, but, if specified in the left and center expressions of (51), would be $\mathbf{x}_j$ to indicate that columns of the **X** matrix are shifted. Note that $\beta$ is respectively a vector and a scalar in the left and centre expressions of (51), and so the sum of the orders of $\beta$ and $\gamma$ equals $\alpha$ in both.

A.2.2.2.4. Three-way and higher-way arrays. In higher-way cases we are faced with new choices. For example, if we want to indicate independent shifting of each fiber (e.g. each column) of a three-way array, $\beta$ is a matrix rather than a vector, with one matrix element for each column in the array. On the other hand, if we want to shift whole slabs of the array at once, $\beta$ is a vector with one element for each slab. Specifying $\gamma$ as well removes any ambiguity.

For example, suppose we independently shift the *JK* columns of an $I \times J \times K$ three-way array $\underline{\mathbf{X}}$. We denote $\beta$ by **S**, a $J \times K$ matrix of shift sizes, and $\gamma$ by $\mathbf{x}_{jk}$, a column vector of $I$ elements taken at levels $j$ and $k$ of the second and third modes of $\underline{\mathbf{X}}$ respectively. Then the shifted version of $\underline{\mathbf{X}}$ is represented as

$$\underset{\mathbf{S}}{\overset{\mathbf{x}_{jk}}{\mathcal{S}}}(\underline{\mathbf{X}}) \tag{52}$$

In (52) we can see that the 'shift unit' **x** is a vector, and its *jk* subscript indicates the $\underline{\mathbf{X}}$ subscripts to be held fixed when determining it (the missing subscript *i* indicates the elements that are to be shifted together, i.e. a column of $\underline{\mathbf{X}}$. The matrix **S** contains a shift size for each of the *JK* shift units in $\underline{\mathbf{X}}$. Note that, unless the context makes it clear that **S** is $J \times K$, one needs to specify the shift unit as well to clarify that it is a column, not a row or 'tube' of $\underline{\mathbf{X}}$ (specifying only the shift unit would allow one to deduce the size of **S**, however).

Now suppose instead that we shift the *K* columns at level *j* by the same amount—in other words, suppose we shift lateral slabs of the array. Then our expression is

$$\underset{\mathbf{s}}{\overset{\mathbf{X}_j}{\mathcal{S}}}(\underline{\mathbf{X}}) \tag{53}$$

where **s** is a *J*-length vector of shift sizes, one for each of the *J* slabs, and $\mathbf{X}_j$ is the $I \times K$ matrix of elements taken from the *j*th lateral slab of $\underline{\mathbf{X}}$ that are shifted together (i.e. the shift unit). Again, explicitly denoting both the shift size and shift unit removes any ambiguity.

In the above examples the shift size vector or matrix is not subscripted, but a subscripted shift vector is often used in this paper when discussing various shifted factor models; such notation implies that the shift vector is part of a shift *matrix*. This occurs when we represent the $(I \times K)$ *j*th lateral

slab of $\underline{\mathbf{X}}$ in shifted factor notation as

$$\mathbf{X}_j = \underset{\mathbf{s}_j}{\mathcal{S}}(\mathbf{A})\langle\mathbf{b}_j\rangle\mathbf{C}' \tag{54}$$

or

$$\mathbf{X}_j = \underset{j}{\mathcal{S}}(\mathbf{A})\langle\mathbf{b}_j\rangle\mathbf{C}' \tag{55}$$

where $\mathbf{A}$ is the $I \times R$ matrix of mode A factor loadings; $\mathbf{b}_j$ is row $j$ from $\mathbf{B}$, the $J \times R$ matrix of mode B factor loadings, and $\langle\mathbf{b}_j\rangle \equiv \mathrm{diag}(\mathbf{b}_j)$ [41]; $\mathbf{C}$ is the $K \times R$ matrix of mode C loadings; and $R$ is the number of factors. Mode A is defined to be the shifted mode, and the shift unit is a column of $\mathbf{A}$; the shift unit is not stated explicitly here, but, if so, would be $\mathbf{a}_r$. The shift size $\mathbf{s}_j$ is an $R$-length vector, one element for each column of $\mathbf{A}$, and the subscript $j$ corresponds to the $\mathbf{X}$ subscript and *not* the column of $\mathbf{A}$. In other words, there are $J$ different $\mathbf{s}$ vectors, one for each slab of $\underline{\mathbf{X}}$, and $\mathbf{s}$ is row $j$ of the $J \times R$ shift matrix $\mathbf{S}$.

## A.3. Algebraic properties of the shift operator

### *A.3.1. Multiple shifts of a given vector*

Considered as an algebraic operation, the shift operation is essentially just addition, except that it is applied to subscript values rather than to variable values. Consequently, shifts of shifted expressions display the same algebraic properties as addition. Like addition, the shift operation is commutative, distributive and associative. For example,

$$\underset{t}{\mathcal{S}}\left(\underset{s}{\mathcal{S}}(\mathbf{X})\right) = \underset{s}{\mathcal{S}}\left(\underset{t}{\mathcal{S}}(\mathbf{X})\right) \tag{56}$$

$$\underset{s}{\mathcal{S}}(\mathbf{x}+\mathbf{y}) = \underset{s}{\mathcal{S}}(\mathbf{x}) + \underset{s}{\mathcal{S}}(\mathbf{y}) \tag{57}$$

$$\begin{aligned}\underset{s}{\mathcal{S}}\left(\underset{t}{\mathcal{S}}\left(\underset{u}{\mathcal{S}}(\mathbf{x})\right)\right) &= \underset{s+t}{\mathcal{S}}\left(\underset{u}{\mathcal{S}}(\mathbf{x})\right)\\ = \underset{s}{\mathcal{S}}\left(\underset{t+u}{\mathcal{S}}(\mathbf{x})\right) &= \underset{s+t+u}{\mathcal{S}}(\mathbf{x})\end{aligned} \tag{58}$$

For any given shift there is an inverse shift, which is of course equivalent to shifting by the negative of the given shift. That is,

$$\mathcal{S}_s^{-1}(\mathbf{x}) = \mathcal{S}_{-s}(\mathbf{x}) \tag{59}$$

There is also an identity shift

$$\mathcal{S}_0(\mathbf{x}) = \mathbf{x} \tag{60}$$

and so we can write

$$\mathcal{S}_{-s}(\mathcal{S}_s(\mathbf{x})) = \mathcal{S}_0(\mathbf{x}) = \mathbf{x} \tag{61}$$

These properties allow the multiple applications of a shift operator to a given vector to be interpreted in a natural way.

### *A.3.2. Shifts of two vectors: vector equality before and after the shifts*

Mathematical statements of equality between two shifted vectors $\mathbf{u}$ and $\mathbf{v}$ (or between a shifted and an unshifted vector) are true or false in the usual sense of vector equality (i.e. true if and only if $u_i = v_i$ for all valid $i$).

However, there are different logical relationships between statements of equality of preshifted vectors and statements of equality of postshifted vectors (i.e. statements involving maintenance of equality under shifting) depending on which model of vector ends one adopts.

Under the 'infinite window' interpretation, vector equality before shifting both implies and is implied by equality after shifting. Thus, when the shifted vectors are treated as unbounded, we can write

$$u = v \leftrightarrow \mathcal{S}(u) = \mathcal{S}(v) \tag{62}$$

Under the 'shrinking window' interpretation, vector equality before shifting implies equality after shifting, but not *vice versa*. Symbolically,

$$u = v \rightarrow \mathcal{S}(u) = \mathcal{S}(v) \tag{63}$$

Finally, under the 'sliding window' interpretation, at least as defined in Section A.1.3.1, there is no implication in either direction:

$$u = v \leftarrow\!\mathrm{X}\!\rightarrow \mathcal{S}(u) = \mathcal{S}(v) \tag{64}$$

However, there are circumstances where it is reasonable to assume (62), i.e. that is, to stipulate the maintenance of equality, even in the 'sliding window' case. This is equivalent to assuming that $\underset{\leftrightarrow}{\mathrm{u}}_{up+i-1} = \underset{\leftrightarrow}{\mathrm{v}}_{vp+i-1}$ for all subscript values $i$ that are defined for $\mathbf{u}$ and $\mathbf{v}$.

## REFERENCES


1. Malinowski ER. *Factor Analysis in Chemistry* (2nd edn). Wiley: New York, 1991.
2. Cattell RB. *The Scientific Use of Factor Analysis in Behavioral and Life Sciences*. Plenum: New York, 1978.
3. Gampp H, Maeder M, Meyer CJ, Zuberbuhler AD. Calculation of equilibrium constants from multiwavelength spectroscopic data—III. *Talanta* 1985; **32**: 1133–1139.
4. Sanchez E, Kowalski BR. Generalized rank annihilation factor analysis. *Anal. Chem*. 1986; **58**: 496–501.
5. Wilson BE, Sanchez E, Kowalski BR. An improved algorithm for the generalized rank annihilation method. *J. Chemometrics* 1989; **3**: 493.
6. Geladi P, Wold S. Local principal component models, rank maps and contextuality for curve resolution and multi-way calibration inference. *Chemometrics Intell. Lab. Syst*. 1987; **2**: 273–281.
7. Karjalainen EJ, Karjalaien UP. Component reconstruction in the primary space of spectra and concentrations—alternating regression and related direct methods. *Anal. Chim. Acta* 1991; **250**: 169–179.
8. Karjalainen EJ, Karjalaien UP. *Data Analysis for Hyphenated Techniques*, vol. 17 of *Data Handling in Science and Technology*. Elsevier: Amsterdam, 1996.
9. Bro R, de Jong S. A fast non-negativity-constrained linear least squares algorithm for use in multi-way algorithms. *J. Chemometrics* 1997; **11**: 393–401.
10. Tauler R, Smilde A, Kowalski BR. Selectivity, local rank, three-way data-analysis and ambiguity in multivariate curve resolution. *J. Chemometrics* 1995; **9**: 31–58.
11. Bro R. *Multi-way Analysis in the Food Industry: Models, Algorithms and Applications*. University of Amsterdam: Amsterdam, 1998 (http://www.models.kvl.dk/users/rasmus/thesis/thesis.html).
12. Eysenck HJ. Criterion analysis—an application of the hypothetico-deductive method to factor analysis. *Psychol. Rev.* 1950; **57**: 38–53.
13. Digman JM. The Procrustes class of factor-analytic transformations. *Multivar. Behav. Anal.* 1967; **2**: 89–94.
14. Harman HH. *Modern Factor Analysis* (3rd edn rev.). University of Chicago Press: Chicago, IL, 1976.
15. Horst P. *Factor Analysis of Data Matrices*. Holt, Rinehart and Winston: New York, 1965.
16. Kaiser HF. The varimax criterion for analytic rotation of factor analysis. *Psychometrika* 1958; **23**: 187–200.
17. Cattell RB. 'Parallel proportional profiles' and other principles for determining the choice of factors by rotation. *Psychometrika* 1944; **9**: 267–283.

18. Gorsuch RL. *Factor Analysis* (2nd edn). Lawrence Erlbaum Associates: Hillsdale, NJ, 1983.
19. Coppi R, Bolasco S (eds). *Multiway Data Analysis*. North-Holland: Amsterdam, 1989.
20. Law HG, Snyder Jr, CW, Hattie JA, McDonald RP (eds). *Research Methods for Multimode Data Analysis*. Praeger: New York, 1984.
21. Harshman RA, Lundy ME. PARAFAC: parallel factor analysis. *Comput. Statist. Data Anal.* 1994; **18**: 39–72.
22. Leurgans S, Ross RT. Multilinear models: applications in spectroscopy. *Statist. Sci.* 1992; **7**: 289–319.
23. Sidiropoulos N, Liu X. PARAFAC methods for blind beamforming—Part I. Identifiability. *IEEE Trans. Signal Process*. In press.
24. Tauler R. Multivariate curve resolution applied to second order data. *Chemometrics Intell. Lab. Syst.* 1995; **30**: 133–146.
25. Bro R, Andersson CA, Kiers HAL. PARAFAC2—Part II: Modeling chromatographic data with retention time shifts. *J. Chemometrics* 1999; **13**: 295–309.
26. Hong S, Harshman RA. Shifted factor analysis. Part III. *N*-way generalization and application. *J. Chemometrics* 2003; **17**: 389–399.
27. Field AS, Graupe D. Topographic component (parallel factor) analysis of multichannel evoked potentials: practical issues in trilinear spatiotemporal decomposition. *Brain Topogr.* 1991; **3**: 407–423.
28. Wulfert F, Kok WT, Smilde AK. Influence of temperature on vibrational spectra and consquences for the predictive ability of multivariate models. *Anal. Chem*. 1998; **70**: 1761–1767.
29. Cattell RB. The structuring of change by P-technique and incremental R-technique. In *Problems in Measuring Change*, Harris CW (ed.). University of Wisconsin Press: Madison, WI, 1963; 167–198.
30. Molenaar PCM. A dynamic factor analysis for the analysis of multivariate time series. *Psychometrika* 1985; **50**: 181–202.
31. Prazen BJ, Synovec RE, Kowalski BR. Standardization of second-order chromatographic/spectroscopic data for optimum chemical analysis. *Anal. Chem*. 1998; **70**: 218–225.
32. Fraga CG, Prazen BJ, Synovec RE. Comprehensive two-dimensional gas chromatography and chemometrics for the high-speed quantitative analysis of aromatic isomers in a jet fuel using the standard addition method and an objective retention time alignment algorithm. *Anal. Chem*. 2000; **72**: 4154–4162.
33. Vogels JTWE, Tas AC, Venekamp J, Van Der Greef J. Partial linear fit: a new NMR spectroscopy preprocessing tool for pattern recognition applications. *J. Chemometrics* 1996; **10**: 425–438.
34. Nielsen N-PV, Carstensen JM, Smedsgaard J. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimized warping. *J. Chromatogr A* 1998; **805**: 17–35.
35. Bylund D, Danielsson R, Malmquist G, Markides KE. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modeling of liquid chromatography–mass spectrometry data. *J. Chromatogr. A* 2002; **961**: 237–244.
36. Westad F, Martens H. Shift and intensity modeling in spectroscopy—general concept and applications. *Chemometrics Intell. Lab. Syst*. 1999; **45**: 361–370.
37. Horn BKP, Schunck BG. Determining optical flow. *Artific. Intell.* 1981; **17**: 185–203.
38. Harshman RA. An index formalism that generalizes the capabilities of matrix notation and algebra to *n*-way arrays. *J. Chemometrics* 2001; **15**: 689–714.
39. 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.
40. Hong S, Harshman RA. Shifted factor analysis. Part II. Algorithms. *J. Chemometrics* 2003; **17**: 379–388.
41. Harshman RA, Hong S. 'Stretch' vs. 'slice' methods for representing three-way structure via matrix notation. *J. Chemometrics* 2002; **16**: 198–205.
42. Harshman RA. Foundations of the Parafac procedure: models and conditions for an 'explanatory' multi-modal factor analysis. *UCLA Working Papers Phonet.* 1970; **16**: 1–84. http://publish.uwo.ca/~harshman/wpppfac0.pdf.
43. Harshman RA. Determination and proof of minimal uniqueness conditions for Parafac1. *UCLA Working Papers Phonet.* 1972; **22**: 111–117. http://publish.uwo.ca/~harshman/wpppfac1.pdf.
44. Hong S. Shifted factor analysis: a test of models and algorithms. *Master's Thesis*, University of Western Ontario, London, ON, 1997.
45. Harshman RA, 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/lawch5.pdf.
46. Carroll JD, Chang JJ. Analysis of individual differences in multidimensional scaling via *N*-way generalization of 'Eckart–Young' decomposition. *Psychometrika* 1970; **35**: 283–319.
47. Kiers HAL. Towards a standardized notation and terminology in multiway analysis. *J. Chemometrics* 2000; **14**: 105–122.
48. Kroonenberg PM. *Three-mode Principal Component Analysis*. DSWO Press: Leiden, 1983.
49. Harshman RA, Lundy ME. Data preprocessing and the extended Parafac model. In *Research Methods for Multimode Data Analysis*, Law HG, Snyder Jr, CW, Hattie JA, McDonald RP (eds). Praeger: New York, 1984; 216–284. http://publish.uwo.ca/~harshman/lawch6.pdf.
50. Kiers HAL, Smilde AK. Constrained three-mode factor analysis as a tool for parameter estimation with second-order instrumental data. *J. Chemometrics* 1998; **12**: 125–147.
51. Kruskal JB. Factor analysis and principal components: bilinear methods. In *International Encyclopedia of Statistics*, Kruskal WH, Tanur JM (eds). Free Press: New York, 1978; 307–330.
52. Windig W, Antalek B. Direct exponential curve resolution (DECRA): a novel application of the generalized rank annihilation method for a single spectral mixture data set with exponentially decaying contribution profiles. *Chemometrics Intell. Lab. Syst*. 1997; **37**: 241–254.
53. Roy R, Kailath T. Estimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust., Speech, Signal Process*. 1989; **37**: 984–995.
54. Windig W, Antalek B, Robbins MJ, Zumbulyadis N, Heckler CE. Applications of the direct exponential curve resolution algorithm (DECRA) to solid state nuclear magnetic resonance and mid-infrared spectra. *J. Chemometrics* 2000; **14**: 213–227.
55. Sidiropolous ND. Generalizing Carathéodory's uniqueness of harmonic parameterization to *N* dimensions. *IEEE Trans. Info. Theory* 2001; **47**: 1687–1690.
56. Jiang T, Sidiropolous ND, ten Berge JMF. Almost sure identifiability of multidimensional harmonic retrieval. *IEEE Trans. Signal Process*. 2001; **49**: 1849–1859.
57. Li T, Sidiropolous ND. Blind digital signal separation using successive interference cancellation iterative least squares. *IEEE Trans. Signal Process*. 2000; **48**: 3146–3152.
58. Wulfert F, Kok WT, de Noord OE, Smilde A. Correction of temperature-induced spectral variation by continuous piecewise direct standardization. *Anal. Chem*. 2000; **72**: 1639–1644.

 