Preprint of article to appear in SIAM Review (June 10, 2008).

Tensor Decompositions and Applications*

Tamara G. Kolda†
Brett W. Bader‡

Abstract. This survey provides an overview of higher-order tensor decompositions, their applications, and available software. A tensor is a multidimensional or N-way array. Decompositions of higher-order tensors (i.e., N-way arrays with $N \geq 3$) have applications in psychometrics, chemometrics, signal processing, numerical linear algebra, computer vision, numerical analysis, data mining, neuroscience, graph analysis, and elsewhere. Two particular tensor decompositions can be considered to be higher-order extensions of the matrix singular value decomposition: CANDECOMP/PARAFAC (CP) decomposes a tensor as a sum of rank-one tensors, and the Tucker decomposition is a higher-order form of principal component analysis. There are many other tensor decompositions, including INDSCAL, PARAFAC2, CANDELINC, DEDICOM, and PARATUCK2 as well as nonnegative variants of all of the above. The N-way Toolbox, Tensor Toolbox, and Multilinear Engine are examples of software packages for working with tensors.

Key words. tensor decompositions, multiway arrays, multilinear algebra, parallel factors (PARAFAC), canonical decomposition (CANDECOMP), higher-order principal components analysis (Tucker), higher-order singular value decomposition (HOSVD)

AMS subject classifications. 15A69, 65F99

1. Introduction.

A *tensor* is a multidimensional array. More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system. This notion of tensors is not to be confused with tensors in physics and engineering (such as stress tensors) [175], which are generally referred to as *tensor fields* in mathematics [69]. A third-order tensor has three indices as shown in Figure 1.1. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors.

The goal of this survey is to provide an overview of higher-order tensors and their decompositions. Though there has been active research on tensor decompositions and models (i.e., decompositions applied to data arrays for extracting and explaining their properties) for four decades, very little of this work has been published in applied mathematics journals. Therefore, we wish to bring this research to the attention of SIAM readers.

Tensor decompositions originated with Hitchcock in 1927 [105, 106], and the idea of a multi-way model is attributed to Cattell in 1944 [40, 41]. These concepts received scant attention until the work of Tucker in the 1960s [224, 225, 226] and Carroll and Chang [38] and Harshman [90] in 1970, all of which appeared in psychometrics

*This work was funded by Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

†Mathematics, Informatics, and Decisions Sciences Department, Sandia National Laboratories, Livermore, CA 94551-9159. Email: tgkolda@sandia.gov.

‡Computer Science and Informatics Department, Sandia National Laboratories, Albuquerque, NM 87185-1318. Email: bwbader@sandia.gov.

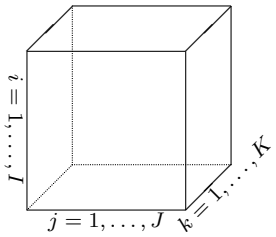


Fig. 1.1: A third-order tensor: $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$

literature. Appellof and Davidson [13] are generally credited as being the first to use tensor decompositions (in 1981) in chemometrics, and tensors have since become extremely popular in that field [103, 201, 27, 28, 31, 152, 241, 121, 12, 9, 29], even spawning a book in 2004 [200]. In parallel to the developments in psychometrics and chemometrics, there was a great deal of interest in decompositions of bilinear forms in the field of algebraic complexity; see, e.g., Knuth [130, §4.6.4]. The most interesting example of this is Strassen matrix multiplication, which is an application of a decomposition of a $4 \times 4 \times 4$ tensor to describe 2×2 matrix multiplication [208, 141, 147, 24].

In the last ten years, interest in tensor decompositions has expanded to other fields. Examples include signal processing [62, 196, 47, 43, 43, 68, 80, 173, 60, 61], numerical linear algebra [87, 63, 64, 132, 244, 133, 149], computer vision [229, 230, 231, 190, 236, 237, 193, 102, 235], numerical analysis [22, 108, 23, 89, 114, 88, 115], data mining [190, 4, 157, 211, 5, 209, 210, 44, 14], graph analysis [136, 135, 15], neuroscience [20, 163, 165, 167, 170, 168, 169, 2, 3, 70, 71] and more. Several surveys have been written already in other fields [138, 52, 104, 27, 28, 47, 129, 78, 48, 200, 69, 29, 6, 184], and a book has appeared very recently on multiway data analysis [139]. Moreover, there are several software packages for working with tensors [179, 11, 146, 85, 16, 17, 18, 239, 243].

Wherever possible, the titles in the references section of this review are hyperlinked to either the publisher web page for the paper or the author's version. Many older papers are now available online in PDF format. We also direct the reader to P. Kroonenberg's *three-mode bibliography*[1], which includes several out-of-print books and theses (including his own [138]). Likewise, R. Harshman's web site[2] has many hard-to-find papers, including his original 1970 PARAFAC paper [90] and Kruskal's 1989 paper [143], which is now out of print.

The remainder of this review is organized as follows. Section 2 describes the notation and common operations used throughout the review; additionally, we provide pointers to other papers that discuss notation. Both the CANDECOMP/PARAFAC (CP) [38, 90] and Tucker [226] tensor decompositions can be considered higher-order generalization of the matrix singular value decomposition (SVD) and principal component analysis (PCA). In §3, we discuss the CP decomposition, its connection to tensor rank and tensor border rank, conditions for uniqueness, algorithms and computational issues, and applications. The Tucker decomposition is covered in §4, where we discuss its relationship to compression, the notion of n-rank, algorithms and computational issues, and applications. Section 5 covers other decompositions, including INDSCAL

[1] `http://three-mode.leidenuniv.nl/bibliogr/bibliogr.htm`
[2] `http://publish.uwo.ca/~harshman/`

Preprint of article to appear in SIAM Review (June 10, 2008).

[38], PARAFAC2 [92], CANDELINC [39], DEDICOM [93], and PARATUCK2 [100], and their applications. Section 6 provides information about software for tensor computations. We summarize our findings in §7.

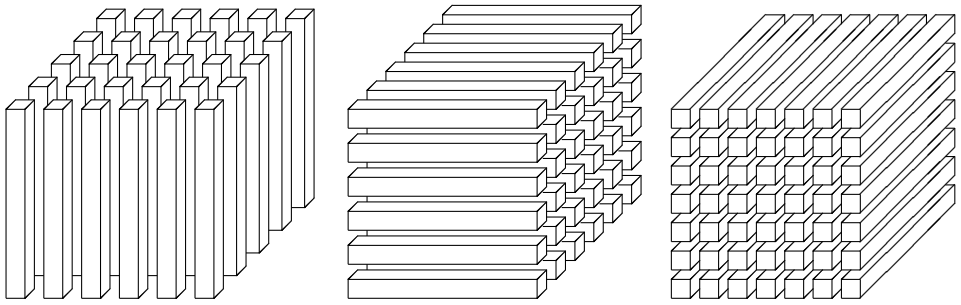
2. Notation and preliminaries. In this review, we have tried to remain as consistent as possible with terminology that would be familiar to applied mathematicians and the terminology of previous publications in the area of tensor decompositions. The notation used here is very similar to that proposed by Kiers [122]. Other standards have been proposed as well; see Harshman [94] and Harshman and Hong [96].

The *order* of a tensor is the number of dimensions, also known as ways or modes.[3] Vectors (tensors of order one) are denoted by boldface lowercase letters, e.g., $\mathbf{a}$. Matrices (tensors of order two) are denoted by boldface capital letters, e.g., $\mathbf{A}$. Higher-order tensors (order three or higher) are denoted by boldface Euler script letters, e.g., $\boldsymbol{\mathcal{X}}$. Scalars are denoted by lowercase letters, e.g., a.

The ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i,j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, and element (i,j,k) of a third-order tensor $\boldsymbol{\mathcal{X}}$ is denoted by x_{ijk}. Indices typically range from 1 to their capital version, e.g., $i = 1, \ldots, I$. The nth element in a sequence is denoted by a superscript in parentheses, e.g., $\mathbf{A}^{(n)}$ denotes the nth matrix in a sequence.

Subarrays are formed when a subset of the indices is fixed. For matrices, these are the rows and columns. A colon is used to indicate all elements of a mode. Thus, the jth column of $\mathbf{A}$ is denoted by $\mathbf{a}_{:j}$, and the ith row of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_{i:}$. Alternatively, the jth column of a matrix, $\mathbf{a}_{:j}$, may be denoted more compactly as $\mathbf{a}_j$.

Fibers are the higher order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensors have column, row, and tube fibers, denoted by $\mathbf{x}_{:jk}$, $\mathbf{x}_{i:k}$, and $\mathbf{x}_{ij:}$, respectively; see Figure 2.1. When extracted from the tensor, fibers are always assumed to be oriented as column vectors.



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

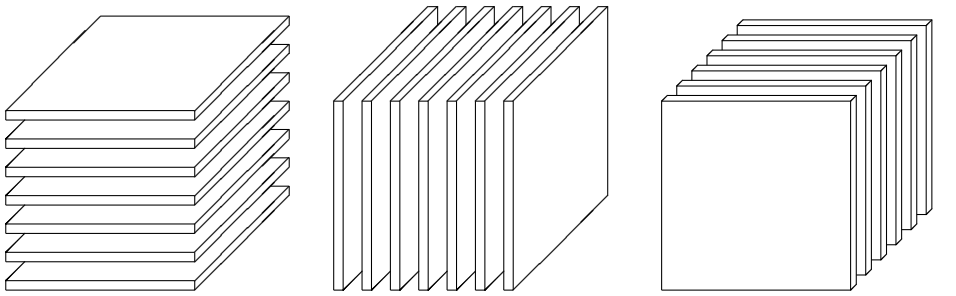
Fig. 2.1: Fibers of a 3rd-order tensor.

Slices are two-dimensional sections of a tensor, defined by fixing all but two indices. Figure 2.2 shows the horizontal, lateral, and frontal slides of a third-order tensor $\boldsymbol{\mathcal{X}}$, denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$, and $\mathbf{X}_{::k}$, respectively. Alternatively, the kth frontal

[3] In some fields, the order of the tensor is referred to as the *rank* of the tensor. In much of the literature and this review, however, the term rank means something quite different; see §3.1.

Preprint of article to appear in SIAM Review (June 10, 2008).

slice of a third-order tensor, $\mathbf{X}_{:\!:k}$, may be denoted more compactly as $\mathbf{X}_k$.



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{:\!:k}$ (or $\mathbf{X}_k$)

Fig. 2.2: Slices of a 3rd-order tensor.

The *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the square root of the sum of the squares of all its elements, i.e.,

$$\| \mathcal{X} \| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N}^2}.$$

This is analogous to the matrix Frobenius norm, which is denoted $\| \mathbf{A} \|$ for a matrix $\mathbf{A}$. The *inner product* of two same-sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the sum of the products of their entries, i.e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}.$$

It follows immediately that $\langle \mathcal{X}, \mathcal{X} \rangle = \| \mathcal{X} \|^2$.

2.1. Rank-one tensors. An N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is *rank one* if it can be written as the outer product of N vectors, i.e.,

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}.$$

The symbol "$\circ$" represents the vector outer product. This means that each element of the tensor is the product of the corresponding vector elements:

$$x_{i_1 i_2 \cdots i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \cdots a_{i_N}^{(N)} \quad \text{for all } 1 \leq i_n \leq I_n.$$

Figure 2.3 illustrates $\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$, a third-order rank-one tensor.

2.2. Symmetry and tensors. A tensor is called *cubical* if every mode is the same size, i.e., $\mathcal{X} \in \mathbb{R}^{I \times I \times I \times \cdots \times I}$ [49]. A cubical tensor is called *supersymmetric* (though this term is challenged by Comon et al. [49] who instead prefer just "symmetric") if its elements remain constant under any permutation of the indices. For instance, a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times I}$ is supersymmetric if

$$x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji} \quad \text{for all } i, j, k = 1, \ldots, I.$$

Preprint of article to appear in SIAM Review (June 10, 2008).

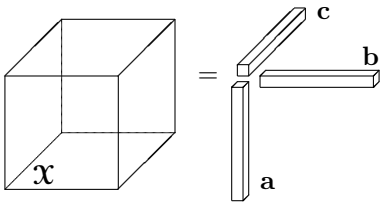


Fig. 2.3: Rank-one third-order tensor, $\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$. The (i, j, k) element of $\mathcal{X}$ is given by $x_{ijk} = a_i b_j c_k$.

Tensors can be (partially) *symmetric* in two or more modes as well. For example, a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ is symmetric in modes one and two if all its frontal slices are symmetric, i.e.,

$$\mathbf{X}_k = \mathbf{X}_k^{\mathsf{T}} \quad \text{for all } k = 1, \ldots, K.$$

Analysis of supersymmetric tensors, which can be shown to be bijectively related to homogeneous polynomials, predates even the work of Hitchcock [106, 105], which was mentioned in the introduction; see [50, 49] for details.

2.3. Diagonal tensors. A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is *diagonal* if $x_{i_1 i_2 \cdots i_N} \neq 0$ only if $i_1 = i_2 = \cdots = i_N$. Figure 2.4 illustrates a cubical tensor with ones along the superdiagonal.

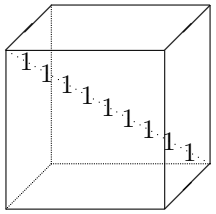


Fig. 2.4: Three-way tensor of size $I \times I \times I$ with ones along the superdiagonal.

2.4. Matricization: transforming a tensor into a matrix. *Matricization*, also known as *unfolding* or *flattening*, is the process of reordering the elements of an N-way array into a matrix. For instance, a $2 \times 3 \times 4$ tensor can be arranged as a 6×4 matrix or a 3×8 matrix, and so on. In this review, we consider only the special case of mode-n matricization because it is the only form relevant to our discussion. A more general treatment of matricization can be found in Kolda [134]. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting matrix. Though conceptually simple, the formal notation is clunky. Tensor element $(i_1, i_2, \ldots, i_N)$ maps to matrix element (i_n, j) where

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k, \quad \text{with} \quad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m.$$

Preprint of article to appear in SIAM Review (June 10, 2008).

The concept is easier to understand using an example. Let the frontal slices of $\mathcal{X} \in \mathbb{R}^{3\times4\times2}$ be

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}. \tag{2.1}$$

Then, the three mode-n unfoldings are:

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix},$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix},$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}.$$

Different papers sometimes use different ordering of the columns for the mode-n unfolding; compare the above with [63] and [122]. In general, the specific permutation of columns is not important so long as it is consistent across related calculations; see [134] for further details.

Lastly, we note that it is also possible to vectorize a tensor. Once again the ordering of the elements is not important so long as it is consistent. In the example above, the vectorized version is:

$$\operatorname{vec}(\mathcal{X}) = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 24 \end{bmatrix}.$$

2.5. Tensor multiplication: the n-mode product. Tensors can be multiplied together, though obviously the notation and symbols for this are much more complex than for matrices. For a full treatment of tensor multiplication see, e.g., Bader and Kolda [16]. Here we consider only the tensor n-mode product, i.e., multiplying a tensor by a matrix (or a vector) in mode n.

The *n-mode (matrix) product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J\times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$ and is of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. Elementwise, we have

$$(\mathcal{X} \times_n \mathbf{U})_{i_1\cdots i_{n-1} j\, i_{n+1}\cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N}\, u_{j i_n}.$$

Each mode-n fiber is multiplied by the matrix $\mathbf{U}$. The idea can also be expressed in terms of unfolded tensors:

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} \quad \Leftrightarrow \quad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}.$$

The n-mode product of a tensor with a matrix is related to a change of basis in the case when a tensor defines a multilinear operator.

Preprint of article to appear in SIAM Review (June 10, 2008).

As an example, let $\mathcal{X}$ be the tensor defined above in (2.1) and let $\mathbf{U} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$. Then, the product $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U} \in \mathbb{R}^{2\times 4\times 2}$ is

$$\mathbf{Y}_1 = \begin{bmatrix} 22 & 49 & 76 & 103 \\ 28 & 64 & 100 & 136 \end{bmatrix}, \quad \mathbf{Y}_2 = \begin{bmatrix} 130 & 157 & 184 & 211 \\ 172 & 208 & 244 & 280 \end{bmatrix}.$$

A few facts regarding n-mode matrix products are in order. For distinct modes in a series of multiplications, the order of the multiplication is irrelevant, i.e.,

$$\mathcal{X} \times_m \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n \mathbf{B} \times_m \mathbf{A} \quad (m \neq n).$$

If the modes are the same, then

$$\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n (\mathbf{BA}).$$

The *n-mode (vector) product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathcal{X} \,\bar{\times}_n\, \mathbf{v}$. The result is of order $N-1$, i.e., the size is $I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N$. Elementwise,

$$(\mathcal{X} \,\bar{\times}_n\, \mathbf{v})_{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N}\, v_{i_n}.$$

The idea is to compute the inner product of each mode-n fiber with the vector $\mathbf{v}$.

For example, let $\mathcal{X}$ be as given in (2.1) and define $\mathbf{v} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}^{\mathsf{T}}$. Then

$$\mathcal{X} \,\bar{\times}_2\, \mathbf{v} = \begin{bmatrix} 70 & 190 \\ 80 & 200 \\ 90 & 210 \end{bmatrix}.$$

When it comes to mode-n vector multiplication, precedence matters because the order of the intermediate results change. In other words,

$$\mathcal{X} \,\bar{\times}_m\, \mathbf{a} \,\bar{\times}_n\, \mathbf{b} = (\mathcal{X} \,\bar{\times}_m\, \mathbf{a}) \,\bar{\times}_{n-1}\, \mathbf{b} = (\mathcal{X} \,\bar{\times}_n\, \mathbf{b}) \,\bar{\times}_m\, \mathbf{a} \quad \text{for} \quad m < n.$$

See [16] for further discussion of concepts in this subsection.

2.6. Matrix Kronecker, Khatri-Rao, and Hadamard products. Several matrix products are important in the sections that follow, so we briefly define them here.

The *Kronecker product* of matrices $\mathbf{A} \in \mathbb{R}^{I\times J}$ and $\mathbf{B} \in \mathbb{R}^{K\times L}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. The result is a matrix of size $(IK) \times (JL)$ and defined by

$$\begin{aligned} \mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_1 \otimes \mathbf{b}_2 & \mathbf{a}_1 \otimes \mathbf{b}_3 & \cdots & \mathbf{a}_J \otimes \mathbf{b}_{L-1} & \mathbf{a}_J \otimes \mathbf{b}_L \end{bmatrix}. \end{aligned}$$

The *Khatri-Rao product* [200] is the "matching columnwise" Kronecker product. Given matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$. The result is a matrix of size $(IJ) \times K$ and defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_K \otimes \mathbf{b}_K \end{bmatrix}.$$

Preprint of article to appear in SIAM Review (June 10, 2008).

If $\mathbf{a}$ and $\mathbf{b}$ are vectors, then the Khatri-Rao and Kronecker products are identical, i.e., $\mathbf{a} \otimes \mathbf{b} = \mathbf{a} \odot \mathbf{b}$.

The *Hadamard product* is the elementwise matrix product. Given matrices $\mathbf{A}$ and $\mathbf{B}$, both of size $I \times J$, their Hadamard product is denoted by $\mathbf{A} * \mathbf{B}$. The result is also size $I \times J$ and defined by

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix}.$$

These matrix products have properties [227, 200] that will prove useful in our discussions:

$$\begin{aligned} (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= \mathbf{AC} \otimes \mathbf{BD}, \\ (\mathbf{A} \otimes \mathbf{B})^{\dagger} &= \mathbf{A}^{\dagger} \otimes \mathbf{B}^{\dagger}, \\ \mathbf{A} \odot \mathbf{B} \odot \mathbf{C} &= (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}), \\ (\mathbf{A} \odot \mathbf{B})^{\mathsf{T}}(\mathbf{A} \odot \mathbf{B}) &= \mathbf{A}^{\mathsf{T}}\mathbf{A} * \mathbf{B}^{\mathsf{T}}\mathbf{B}, \\ (\mathbf{A} \odot \mathbf{B})^{\dagger} &= ((\mathbf{A}^{\mathsf{T}}\mathbf{A}) * (\mathbf{B}^{\mathsf{T}}\mathbf{B}))^{\dagger}(\mathbf{A} \odot \mathbf{B})^{\mathsf{T}}. \end{aligned} \tag{2.2}$$

Here $\mathbf{A}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{A}$ [84].

As an example of the utility of the Kronecker product, consider the following. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathbf{A}^{(n)} \in \mathbb{R}^{J_n \times I_n}$ for all $n \in \{1, \ldots, N\}$. Then, for any $n \in \{1, \ldots, N\}$ we have

$$\begin{aligned} \mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \quad &\Leftrightarrow \\ \mathbf{Y}_{(n)} = \mathbf{A}^{(n)}\mathbf{X}_{(n)} \left(\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}\right)^{\mathsf{T}}. \end{aligned}$$

See [134] for a proof of this property.

3. Tensor rank and the CANDECOMP/PARAFAC decomposition. In 1927, Hitchcock [105, 106] proposed the idea of the polyadic form of a tensor, i.e., expressing a tensor as the sum of a finite number of rank-one tensors; and in 1944 Cattell [40, 41] proposed ideas for parallel proportional analysis and the idea of multiple axes for analysis (circumstances, objects, and features). The concept finally became popular after its third introduction, in 1970 to the psychometrics community, in the form of CANDECOMP (canonical decomposition) by Carroll and Chang [38] and PARAFAC (parallel factors) by Harshman [90]. We refer to the CANDECOMP/PARAFAC decomposition as CP, per Kiers [122]. Möcks [166] independently discovered CP in the context of brain imaging and called it the Topographic Components Model. Table 3.1 summarizes the different names for the CP decomposition.

The CP decomposition factorizes a tensor into a sum of component rank-one tensors. For example, given a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we wish to write it as

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \tag{3.1}$$

Preprint of article to appear in SIAM Review (June 10, 2008).

Name	Proposed by
Polyadic Form of a Tensor	Hitchcock, 1927 [105]
PARAFAC (Parallel Factors)	Harshman, 1970 [90]
CANDECOMP or CAND (Canonical decomposition)	Carroll and Chang, 1970 [38]
Topographic Components Model	Möcks, 1988 [166]
CP (CANDECOMP/PARAFAC)	Kiers, 2000 [122]

Table 3.1: Some of the many names for the CP decomposition.

where R is a positive integer, and $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$, for $r = 1, \ldots, R$. Elementwise, (3.1) is written as

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir}\, b_{jr}\, c_{kr}, \text{ for } i = 1, \ldots, I,\ j = 1, \ldots, J,\ k = 1, \ldots, K.$$

This is illustrated in Figure 3.1.

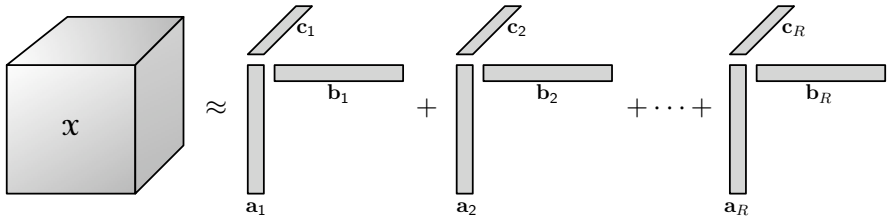


Fig. 3.1: CP decomposition of a three-way array.

The *factor matrices* refer to the combination of the vectors from the rank-one components, i.e., $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_R\end{bmatrix}$ and likewise for $\mathbf{B}$ and $\mathbf{C}$. Using these definitions, (3.1) may be written in matricized form (one per mode; see §2.4):

$$\begin{aligned} \mathbf{X}_{(1)} &\approx \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}, \\ \mathbf{X}_{(2)} &\approx \mathbf{B}(\mathbf{C} \odot \mathbf{A})^{\mathsf{T}}, \\ \mathbf{X}_{(3)} &\approx \mathbf{C}(\mathbf{B} \odot \mathbf{A})^{\mathsf{T}}. \end{aligned} \tag{3.2}$$

Recall that $\odot$ denotes the Khatri-Rao product from §2.6. The three-way model is sometimes written in terms of the frontal slices of $\mathcal{X}$ (see Figure 2.2):

$$\mathbf{X}_k \approx \mathbf{A}\mathbf{D}^{(k)}\mathbf{B}^{\mathsf{T}} \quad \text{where} \quad \mathbf{D}^{(k)} \equiv \operatorname{diag}(\mathbf{c}_{k:}) \quad \text{for} \quad k = 1, \ldots, K.$$

Analogous equations can be written for the horizontal and lateral slices. In general, though, slice-wise expressions do not easily extend beyond three dimensions. Following Kolda [134] (see also Kruskal [141]), the CP model can be concisely expressed as

$$\mathcal{X} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

It is often useful to assume that the columns of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are normalized to length one with the weights absorbed into the vector $\boldsymbol{\lambda} \in \mathbb{R}^R$ so that

$$\mathcal{X} \approx [\![\boldsymbol{\lambda} \,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{3.3}$$

We have focused on the three-way case because it is widely applicable and sufficient for many needs. For a general Nth-order tensor, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the CP decomposition is

$$\mathcal{X} \approx [\![\boldsymbol{\lambda} \,; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \equiv \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}.$$

where $\boldsymbol{\lambda} \in \mathbb{R}^R$ and $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$. In this case, the mode-n matricized version is given by

$$\mathbf{X}_{(n)} \approx \mathbf{A}^{(n)} \boldsymbol{\Lambda} (\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})^{\mathsf{T}},$$

where $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda})$.

3.1. Tensor rank. The *rank* of a tensor $\mathcal{X}$, denoted $\operatorname{rank}(\mathcal{X})$, is defined as the smallest number of rank-one tensors (see §2.1) that generate $\mathcal{X}$ as their sum [105, 141]. In other words, this is the smallest number of components in an exact CP decomposition, where "exact" means that there is equality in (3.1). Hitchcock [105] first proposed this definition of rank in 1927, and Kruskal [141] did so independently 50 years later. For the perspective of tensor rank from an algebraic complexity point-of-view, see [107, 37, 130, 24] and references therein. An exact CP decomposition with $R = \operatorname{rank}(\mathcal{X})$ components is called the *rank decomposition*.

The definition of tensor rank is an exact analogue to the definition of matrix rank, but the properties of matrix and tensor ranks are quite different. One difference is that the rank of a real-valued tensor may actually be different over $\mathbb{R}$ and $\mathbb{C}$. For example, let $\mathcal{X}$ be a tensor whose frontal slices are defined by

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

This is a tensor of rank-3 over $\mathbb{R}$ and rank-2 over $\mathbb{C}$. The rank decomposition over $\mathbb{R}$ is $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}.$$

Whereas the rank decomposition over $\mathbb{C}$ has the following factor matrices instead:

$$\mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}, \quad \mathbf{B} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}, \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}.$$

This example comes from Kruskal [142], and the proof that it is rank-3 over $\mathbb{R}$ and the methodology for computing the factors can be found in Ten Berge [212]. See also Kruskal [143] for further discussion.

Another major difference between matrix and tensor rank is that (except in special cases such as the example above) there is no straightforward algorithm to determine

Preprint of article to appear in SIAM Review (June 10, 2008).

the rank of a specific given tensor; in fact, the problem is NP-hard [101]. Kruskal [143] cites an example of a particular $9 \times 9 \times 9$ tensor whose rank is only known to be bounded between 18 and 23 (recent work by Comon et al. [51] conjectures that the rank is 19 or 20). In practice, the rank of a tensor is determined numerically by fitting various rank-R CP models; see §3.4.

Another peculiarity of tensors has to do with maximum and typical ranks. The *maximum rank* is defined as the largest attainable rank, whereas the *typical rank* is any rank that occurs with probability greater than zero (i.e., on a set with positive Lebesgue measure). For the collection of $I \times J$ matrices, the maximum and typical ranks are identical and equal to $\min\{I, J\}$. For tensors, the two ranks may be different. Moreover, over $\mathbb{R}$, there may be more than one typical rank whereas there is always only one typical rank over $\mathbb{C}$. Kruskal [143] discusses the case of $2 \times 2 \times 2$ tensors which have typical ranks of two and three over $\mathbb{R}$. In fact, Monte Carlo experiments (which randomly draw each entry of the tensor from a normal distribution with mean zero and standard deviation one) reveal that the set of $2 \times 2 \times 2$ tensors of rank two fills about 79% of the space while those of rank three fill 21%. Rank-one tensors are possible but occur with zero probability. See also the concept of *border rank* discussed in §3.3.

For a general third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, only the following weak upper bound on its maximum rank is known [143]:

$$\operatorname{rank}(\mathcal{X}) \leq \min\{IJ, IK, JK\}.$$

Table 3.2 shows known maximum ranks for tensors of specific sizes. The most general result is for third-order tensors with only two slices.

Tensor Size	**Maximum Rank**	**Citation**
$I \times J \times 2$	$\min\{I, J\} + \min\{I, J, \lfloor \max\{I, J\}/2 \} \rfloor\}$	[109, 143]
$3 \times 3 \times 3$	5	[143]

Table 3.2: Maximum ranks over $\mathbb{R}$ for three-way tensors.

Table 3.3 shows some known formulas for the typical ranks of certain three-way tensors over $\mathbb{R}$. For general $I \times J \times K$ (or higher order) tensors, recall that the ordering of the modes does not affect the rank, i.e., the rank is constant under permutation. Kruskal [143] discussed the case of $2 \times 2 \times 2$ tensors having typical ranks of both two and three and gives a diagnostic polynomial to determine the rank of a given tensor. Ten Berge [212] later extended this result (and the diagnostic polynomial) to show that $I \times I \times 2$ tensors have typical ranks of I and $I+1$. Ten Berge and Kiers [216] fully characterize the set of all $I \times J \times 2$ arrays and further contrast the difference between rank over $\mathbb{R}$ and $\mathbb{C}$. The typical rank over $\mathbb{R}$ of an $I \times J \times 2$ tensor is $\min\{I, 2J\}$ when $I > J$, and the same holds true over $\mathbb{C}$. However, the typical rank(s) for an $I \times J \times 2$ tensor with $I = J$ (i.e., a tensor of size $I \times I \times 2$) is $\{I, I + 1\}$ over $\mathbb{R}$ but I over $\mathbb{C}$. For general three-way arrays, Ten Berge [213] classifies their ranks over $\mathbb{R}$ according to the sizes of I, J and K, as follows: A tensor with $I > J > K$ is called "very tall" when $I > JK$; "tall" when $JK - J < I < JK$, and "compact" when $I < JK - J$. Very tall tensors trivially have maximal and typical rank JK [213]. Tall tensors are treated in [213] (and [216] for $I \times J \times 2$ arrays). Little is known about the typical rank of compact tensors except for when $I = JK - J$ [213]. Recent results by Comon et al. [51] provide typical ranks for a larger range of tensor sizes.

Tensor Size	Typical Rank	Citation
$2 \times 2 \times 2$	$\{2, 3\}$	[143]
$3 \times 3 \times 2$	$\{3, 4\}$	[142, 212]
$5 \times 3 \times 3$	$\{5, 6\}$	[214]
$I \times J \times 2$ with $I \geq 2J$ (very tall)	$2J$	[216]
$I \times J \times 2$ with $J < I < 2J$ (tall)	I	[216]
$I \times I \times 2$ (compact)	$\{I, I+1\}$	[212, 216]
$I \times J \times K$ with $I \geq JK$ (very tall)	JK	[213]
$I \times J \times K$ with $JK - J < I < JK$ (tall)	I	[213]
$I \times J \times K$ with $I = JK - J$ (compact)	$\{I, I+1\}$	[213]

Table 3.3: Typical rank over $\mathbb{R}$ for three-way tensors.

The situation changes somewhat when we restrict the tensors in some way. Ten Berge, Sidiropoulos, and Rocci [219] consider the case where the tensor is symmetric in two modes. Without loss of generality, assume that the frontal slices are symmetric; see §2.2. The results are presented in Table 3.4. It is interesting to compare the results for tensors with and without symmetric slices as is done in Table 3.5. In some cases, e.g., $I \times I \times 2$, the typical rank is the same in either case. But in other cases, e.g., $I \times 2 \times 2$, the typical rank differs.

Tensor Size	Typical Rank
$I \times I \times 2$	$\{I, I+1\}$
$3 \times 3 \times 3$	4
$3 \times 3 \times 4$	$\{4, 5\}$
$3 \times 3 \times 5$	$\{5, 6\}$
$I \times I \times K$ with $K \geq I(I+1)/2$	$I(I+1)/2$

Table 3.4: Typical rank over $\mathbb{R}$ for three-way tensors with symmetric frontal slices [219].

Tensor Size	Typical Rank with Symmetry	Typical Rank without Symmetry
$I \times I \times 2$ (compact)	$\{I, I+1\}$	$\{I, I+1\}$
$2 \times 3 \times 3$ (compact)	$\{3, 4\}$	$\{3, 4\}$
$3 \times 2 \times 2$ (tall)	3	3
$I \times 2 \times 2$ with $I \geq 4$ (very tall)	3	4
$I \times 3 \times 3$ with $I = 6, 7, 8$ (tall)	6	I
$9 \times 3 \times 3$ (very tall)	6	9

Table 3.5: Comparison of typical ranks over $\mathbb{R}$ for three-way tensors with and without symmetric frontal slices [219].

Comon, Golub, Lim, and Mourrain [49] have recently investigated the special case of supersymmetric tensors (see §2.2) over $\mathbb{C}$. Let $\mathcal{X} \in \mathbb{C}^{I \times I \times \cdots \times I}$ be a supersymmetric tensor of order N. Define the *symmetric rank* (over $\mathbb{C}$) of $\mathcal{X}$ to be

$$\operatorname{rank}_S(\mathcal{X}) = \min \left\{ R \ : \ \mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{a}_r \circ \cdots \circ \mathbf{a}_r \text{ where } \mathbf{A} \in \mathbb{C}^{I \times R} \right\},$$

Preprint of article to appear in SIAM Review (June 10, 2008).

i.e., the minimum number of symmetric rank-one factors. Comon et al. show that, with probability one,

$$\operatorname{rank}_S(\mathcal{X}) = \left\lceil \frac{\binom{I+N-1}{N}}{I} \right\rceil,$$

except for when $(N, I) \in \{(3,5),(4,3),(4,4),(4,5)\}$ in which case it should be increased by one. The result is due to Alexander and Hirschowitz [7, 49].

3.2. Uniqueness. An interesting property of higher-order tensors is that their rank decompositions are oftentimes unique, whereas matrix decompositions are not. Sidiropoulos and Bro [199] and Ten Berge [213] provide some history of uniqueness results for CP. The earliest uniqueness result is due to Harshman in 1970 [90], which he in turn credits to Dr. Robert Jennich. Harshman's result is a special case of the more general results presented here. Harshman presented further results in 1972 [91] which are generalized in [143, 212].

Consider the fact that matrix decompositions are not unique. Let $\mathbf{X} \in \mathbb{R}^{I \times J}$ be a matrix of rank R. Then a rank decomposition of this matrix is:

$$\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathsf{T}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r.$$

If the SVD of $\mathbf{X}$ is $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}$, then we can choose $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}$ and $\mathbf{B} = \mathbf{V}$. However, it is equally valid to choose $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}$ and $\mathbf{B} = \mathbf{V}\mathbf{W}$ where $\mathbf{W}$ is some $R \times R$ orthogonal matrix. In other words, we can easily construct two completely different sets of R rank-one matrices that sum to the original matrix. The SVD of a matrix is unique (assuming all the singular values are distinct) only because of the addition of orthogonality constraints (and the diagonal matrix of ordered singular values in the middle).

The CP decomposition, on the other hand, is unique under much weaker conditions. Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ be a three-way tensor of rank R, i.e.,

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \tag{3.4}$$

Uniqueness means that this is the only possible combination of rank-one tensors that sums to $\mathcal{X}$, with the exception of the elementary indeterminacies of scaling and permutation. The permutation indeterminacy refers to the fact that the rank-one component tensors can be reordered arbitrarily, i.e.,

$$\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = [\![\mathbf{A}\mathbf{\Pi}, \mathbf{B}\mathbf{\Pi}, \mathbf{C}\mathbf{\Pi}]\!] \text{ for any } R \times R \text{ permutation matrix } \mathbf{\Pi}.$$

The scaling indeterminacy refers to the fact that we can scale the individual vectors, i.e.,

$$\mathcal{X} = \sum_{r=1}^{R} (\alpha_r \mathbf{a}_r) \circ (\beta_r \mathbf{b}_r) \circ (\gamma_r \mathbf{c}_r),$$

so long as $\alpha_r \beta_r \gamma_r = 1$ for $r = 1, \ldots, R$.

The most general and well-known result on uniqueness is due to Kruskal [141, 143] and depends on the concept of k-rank. The *k-rank* of a matrix $\mathbf{A}$, denoted $k_{\mathbf{A}}$, is defined as the maximum value k such that any k columns are linearly independent [141, 98]. Kruskal's result [141, 143] says that a sufficient condition for uniqueness for the CP decomposition in (3.4) is

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2.$$

Kruskal's result is non-trivial and has been re-proven and analyzed many times over; see, e.g., [199, 218, 111, 206]. Sidiropoulos and Bro [199] recently extended Kruskal's result to N-way tensors. Let $\mathcal{X}$ be an N-way tensor with rank R and suppose that its CP decomposition is

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]. \tag{3.5}$$

Then a sufficient condition for uniqueness is

$$\sum_{n=1}^{N} k_{\mathbf{A}^{(n)}} \geq 2R + (N-1).$$

The previous results provide only sufficient conditions. Ten Berge and Sidiropoulos [218] show that the sufficient condition is also necessary for tensors of rank $R = 2$ and $R = 3$, but not for $R > 3$. Liu and Sidiropoulos [158] considered more general necessary conditions. They show that a necessary condition for uniqueness of the CP decomposition in (3.4) is

$$\min\{\, \operatorname{rank}(\mathbf{A} \odot \mathbf{B}), \operatorname{rank}(\mathbf{A} \odot \mathbf{C}), \operatorname{rank}(\mathbf{B} \odot \mathbf{C}) \,\} = R.$$

More generally, they show that for the N-way case, a necessary condition for uniqueness of the CP decomposition in (3.5) is

$$\min_{n=1,\ldots,N} \operatorname{rank}\left(\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(n-1)} \odot \mathbf{A}^{(n+1)} \odot \cdots \odot \mathbf{A}^{(N)} \right) = R.$$

They further observe that since $\operatorname{rank}(\mathbf{A} \odot \mathbf{B}) \leq \operatorname{rank}(\mathbf{A} \otimes \mathbf{B}) \leq \operatorname{rank}(\mathbf{A}) \cdot \operatorname{rank}(\mathbf{B})$, an even simpler necessary condition is

$$\min_{n=1,\ldots,N} \left(\prod_{\substack{m=1 \\ m \neq n}}^{N} \operatorname{rank}(\mathbf{A}^{(m)}) \right) \geq R.$$

De Lathauwer [55] has looked at methods to determine the rank of a tensor and the question of when a given CP decomposition is deterministically or *generically* (i.e., with probability one) unique. The CP decomposition in (3.4) is generically unique if

$$R \leq K \quad \text{and} \quad R(R-1) \leq I(I-1)J(J-1)/2.$$

Likewise, a 4th-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K \times L}$ of rank R has a CP decomposition that is generically unique if

$$R \leq L \quad \text{and} \quad R(R-1) \leq IJK(3IJK - IJ - IK - JK - I - J - K + 3)/4.$$

Preprint of article to appear in SIAM Review (June 10, 2008).

3.3. Low-rank approximations and the border rank. For matrices, Eckart and Young [76] showed that a best rank-k approximation is given by the leading k factors of the SVD. In other words, let R be the rank of a matrix $\mathbf{A}$ and assume its SVD is given by

$$\mathbf{A} = \sum_{r=1}^{R} \sigma_r \, \mathbf{u}_r \circ \mathbf{v}_r, \quad \text{with } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R > 0.$$

Then a rank-k approximation that minimizes $\| \mathbf{A} - \mathbf{B} \|$ is given by

$$\mathbf{B} = \sum_{r=1}^{k} \sigma_r \, \mathbf{u}_r \circ \mathbf{v}_r.$$

This type of result does not hold true for higher-order tensors. For instance, consider a third-order tensor of rank R with the following CP decomposition:

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

Ideally, summing k of the factors would yield a best rank-k approximation, but that is not the case. Kolda [132] provides an example where the best rank-one approximation of a cubic tensor is not a factor in the best rank-two approximation. A corollary of this fact is that the components of the best rank-k model may not be solved for sequentially—all factors must be found simultaneously; see [200, Example 4.3].

In general, though, the problem is more complex. It is possible that the best rank-k approximation may not even exist. The problem is referred to as one of degeneracy. A tensor is *degenerate* if it may be approximated arbitrarily well by a factorization of lower rank. An example from [180, 69] best illustrates the problem. Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ be a rank-3 tensor defined by

$$\mathcal{X} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_2 + \mathbf{a}_1 \circ \mathbf{b}_2 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_1 \circ \mathbf{c}_1,$$

where $\mathbf{A} \in \mathbb{R}^{I \times 2}$, $\mathbf{B} \in \mathbb{R}^{J \times 2}$, and $\mathbf{C} \in \mathbb{R}^{K \times 2}$, and each has linearly independent columns. This tensor can be approximated arbitrarily closely by a rank-two tensor of the following form:

$$\mathcal{Y} = \alpha \left(\mathbf{a}_1 + \frac{1}{\alpha} \mathbf{a}_2 \right) \circ \left(\mathbf{b}_1 + \frac{1}{\alpha} \mathbf{b}_2 \right) \circ \left(\mathbf{c}_1 + \frac{1}{\alpha} \mathbf{c}_2 \right) - \alpha \, \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1.$$

Specifically,

$$\| \mathcal{X} - \mathcal{Y} \| = \frac{1}{\alpha} \left\| \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_1 \circ \mathbf{c}_2 + \mathbf{a}_1 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \frac{1}{\alpha} \, \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 \right\|,$$

which can be made arbitrarily small. As this example illustrates, it is typical of degeneracy that factors become nearly proportional and that the magnitude of some terms in the decomposition go to infinity but have opposite sign. We note that Paatero [180] also cites independent, unpublished work by Kruskal [142], and De Silva and Lim [69] cite Knuth [130] for this example.

Paatero [180] provides further examples of degeneracy. Kruskal, Harshman, and Lundy [144] also discuss degeneracy and illustrate the idea of a series of lower-rank

Preprint of article to appear in SIAM Review (June 10, 2008).

tensors converging to one of higher rank. Figure 3.2 shows the problem of estimating a rank-three tensor $\mathcal{Y}$ by a rank-two tensor [144]. Here, a sequence $\{\mathcal{X}_k\}$ of rank-two tensors provides increasingly better estimates of $\mathcal{Y}$. Necessarily, the best approximation is on the boundary of the space of rank-two and rank-three tensors. However, since the space of rank-two tensors is not closed, the sequence may converge to a tensor $\mathcal{X}^*$ of rank other than two. The previous example shows a sequence of rank-two tensors converging to rank-three.

Lundy, Harshman, and Kruskal [159] later propose an alternative numerical model that combines CP with a Tucker decomposition in order to better avoid degeneracies. The earliest example of degeneracy is from Bini, Capovani, Lotti, and Romani [25, 69] in 1979, who give an explicit example of a sequence of rank-5 tensors converging to a rank-6 tensor. De Silva and Lim [69] show, moreover, that the set of tensors of a given size that do not have a best rank-k approximation has positive volume (i.e., positive Lebesgue measure) for at least some values of k, so this problem of a lack of a best approximation is not a "rare" event. In related work, Comon et al. [49] showed similar examples for symmetric tensors and symmetric approximations. Stegeman [202] considers the case of $I \times I \times 2$ arrays and proves that *any* tensor with rank $R = I + 1$ does not have a best rank-I approximation. Further, Stegeman [204] considers the case of all tensors of size $I \times J \times 3$ with two typical ranks and shows that in most cases the higher ranks can be approximated arbitrarily well by a lower-rank tensor.

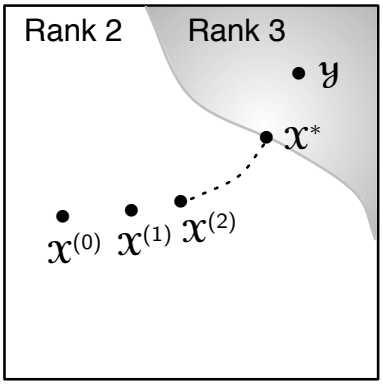


Fig. 3.2: Illustration of a sequence of tensors converging to one of higher rank [144].

In the situation where a best low-rank approximation does not exist, it is useful to consider the concept of *border rank* [25, 24], which is defined as the minimum number of rank-one tensors that are sufficient to approximate the given tensor with arbitrarily small nonzero error. This concept was introduced in 1979 and developed within the algebraic complexity community through the 1980s. Mathematically, border rank is defined as

$$\widetilde{\text{rank}}(\mathcal{X}) = \min\{\, r \mid \text{for any } \epsilon > 0, \text{ there exists a tensor } \mathcal{E} \text{ such that } \|\mathcal{E}\| < \epsilon \text{ and } \text{rank}(\mathcal{X} + \mathcal{E}) = r \,\}. \quad (3.6)$$

An obvious condition is that

$$\widetilde{\text{rank}}(\mathcal{X}) \leq \text{rank}(\mathcal{X}).$$

Preprint of article to appear in SIAM Review (June 10, 2008).

Much of the work on border rank has been done in the context of bilinear forms and matrix multiplication. In particular, Strassen matrix multiplication [208] comes from considering the rank of a particular $4 \times 4 \times 4$ tensor that represents matrix-matrix multiplication for 2×2 matrices. In this case, it can be shown that the rank and border rank of the tensor are both equal to 7 (see, e.g., [147]). The case of 3×3 matrix multiplication corresponds to a $9 \times 9 \times 9$ tensor that has rank between 19 and 23 [145, 26, 24] and a border rank between 14 and 21 [192, 26, 24, 148].

3.4. Computing the CP decomposition. As mentioned previously, there is no finite algorithm for determining the rank of a tensor [143, 101]; consequently, the first issue that arises in computing a CP decomposition is how to choose the number of rank-one components. Most procedures fit multiple CP decompositions with different numbers of components until one is "good". Ideally, if the data are noise-free and we have a procedure for calculating CP with a given number of components, then we can do that computation for $R = 1, 2, 3, \ldots$ number of components and stop at the first value of R that gives a fit of 100%. However, there are many problems with this procedure. We will see below that there is no perfect procedure for fitting CP for a given number of components. Additionally, as we saw in §3.3, some tensors may have approximations of a lower rank that are arbitrarily close in terms of fit, and this does cause problems in practice [164, 186, 180]. When the data are noisy (as is frequently the case), then fit alone cannot determine the rank in any case; instead Bro and Kiers [34] have proposed a consistency diagnostic called CORCONDIA to compare different numbers of components.

Assuming the number of components is fixed, there are many algorithms to compute a CP decomposition. Here we focus on what is today the "workhorse" algorithm for CP: the alternating least squares (ALS) method proposed in the original papers by Carroll and Chang [38] and Harshman [90]. For ease of presentation, we only derive the method in the third-order case, but the full algorithm is presented for an N-way tensor in Figure 3.3.

Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ be a third order tensor. The goal is to compute a CP decomposition with R components that best approximates $\mathcal{X}$, i.e., to find

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\| \quad \text{with} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\boldsymbol{\lambda}\,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \tag{3.7}$$

The alternating least squares approach fixes $\mathbf{B}$ and $\mathbf{C}$ to solve for $\mathbf{A}$, then fixes $\mathbf{A}$ and $\mathbf{C}$ to solve for $\mathbf{B}$, then fixes $\mathbf{A}$ and $\mathbf{B}$ to solve for $\mathbf{C}$, and continues to repeat the entire procedure until some convergence criterion is satisfied.

Having fixed all but one matrix, the problem reduces to a linear least squares problem. For example, suppose that $\mathbf{B}$ and $\mathbf{C}$ are fixed. Then, from (3.2), we can rewrite the above minimization problem in matrix form as

$$\min_{\hat{\mathbf{A}}} \|\mathbf{X}_{(1)} - \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}\|_F,$$

where $\hat{\mathbf{A}} = \mathbf{A} \cdot \operatorname{diag}(\boldsymbol{\lambda})$. The optimal solution is then given by

$$\hat{\mathbf{A}} = \mathbf{X}_{(1)} \left[(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}\right]^{\dagger}.$$

Because the Khatri-Rao product pseudo-inverse has the special form in (2.2), it is common to rewrite the solution as

$$\hat{\mathbf{A}} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^{\mathsf{T}}\mathbf{C} * \mathbf{B}^{\mathsf{T}}\mathbf{B})^{\dagger}.$$

procedure CP-ALS($\mathcal{X}$,R)
  initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$
  repeat
    for $n = 1, \ldots, N$ **do**
      $\mathbf{V} \leftarrow \mathbf{A}^{(1)\mathsf{T}}\mathbf{A}^{(1)} * \cdots * \mathbf{A}^{(n-1)\mathsf{T}}\mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\mathsf{T}}\mathbf{A}^{(n+1)} * \cdots * \mathbf{A}^{(N)\mathsf{T}}\mathbf{A}^{(N)}$
      $\mathbf{A}^{(n)} \leftarrow \mathbf{X}_{(n)}(\mathbf{A}^{(N)} \odot \ldots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \ldots \odot \mathbf{A}^{(1)})\mathbf{V}^{\dagger}$
      normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)
    end for
  until fit ceases to improve or maximum iterations exhausted
  return $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Fig. 3.3: Alternating least squares algorithm to compute a CP decomposition with R components for an Nth order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$.

The advantage of this version of the equation is that we need only calculate the pseudo-inverse of an $R \times R$ matrix rather than a $JK \times R$ matrix; however, this version is not always advised due to the potential of numerical ill-conditioning. Finally, we normalize the columns of $\hat{\mathbf{A}}$ to get $\mathbf{A}$; in other words, let $\lambda_r = \|\hat{\mathbf{a}}_r\|$ and $\mathbf{a}_r = \hat{\mathbf{a}}_r/\lambda_r$ for $r = 1, \ldots, R$.

The full ALS procedure for an N-way tensor is shown in Figure 3.3. It assumes that the number of components, R, of the CP decomposition is specified. The factor matrices can be initialized in any way, such as randomly or by setting

$$\mathbf{A}^{(n)} = R \text{ leading left singular vectors of } \mathbf{X}_{(n)} \quad \text{for} \quad n = 1, \ldots, N.$$

See [28, 200] for more discussion on initializing the method. At each inner iteration, the pseudo-inverse of a matrix $\mathbf{V}$ (see Figure 3.3) must be computed, but it is only of size $R \times R$. The iterations repeat until some combination of stopping conditions is satisfied. Possible stopping conditions include: little or no improvement in the objective function, little or no change in the factor matrices, objective value is at or near zero, and exceeding a predefined maximum number of iterations.

The ALS method is simple to understand and implement, but can take many iterations to converge. Moreover, it is not guaranteed to converge to a global minimum nor even a stationary point of (3.7), only to a solution where the objective function of (3.7) ceases to decrease. The final solution can be heavily dependent on the starting guess as well. Some techniques for improving the efficiency of ALS are discussed in [220, 221]. Several researchers have proposed improving ALS with line searches, including the ELS approach of Rajih and Comon [185] which adds a line search after each major iteration that updates all component matrices simultaneously based on the standard ALS search directions; see also [177]. Navasca et al. [176] propose using Tikhonov regularization on the ALS subproblems.

Two recent surveys summarize other options. In one survey, Faber, Bro, and Hopke [78] compare ALS with six different methods, none of which is better than ALS in terms of quality of solution, though the alternating slicewise diagonalization (ASD) method [110] is acknowledged as a viable alternative when computation time is paramount. In another survey, Tomasi and Bro [223] compare ALS and ASD to four other methods plus three variants that apply Tucker-based compression (see §4 and §5.3) and then compute a CP decomposition of the reduced array; see [30]. In this comparison, damped Gauss-Newton (dGN) and a variant called PMF3 by Paatero [178] are included. Both dGN and PMF3 optimize all factor matrices simultaneously.

Preprint of article to appear in SIAM Review (June 10, 2008).

In contrast to the results of the previous survey, ASD is deemed to be inferior to other alternating-type methods. Derivative-based methods (dGN and PMF3) are generally superior to ALS in terms of their convergence properties but are more expensive in both memory and time. We expect many more developments in this area as more sophisticated optimization techniques are developed for computing the CP model.

Sanchez and Kolwalski [189] propose a method for reducing the CP fitting problem in the three-way case to a generalized eigenvalue problem, which works when the first two factor matrices are full rank. De Lathauwer, De Moor, and Vandewalle [66] cast CP as a simultaneous generalized Schur decomposition (SGSD) if, for a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, $\text{rank}(\mathcal{X}) \leq \min\{I, J\}$ and the matrix rank of the unfolded tensor $\mathbf{X}_{(3)}$ is greater than or equal to 2. The SGSD approach has been applied to overcoming the problem of degeneracy [205]; see also [137]. More recently, De Lathauwer [55] developed a method based on simultaneous matrix diagonalization in the case that, for an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\max\{I_n : n = 1, \ldots, N\} \geq \text{rank}(\mathcal{X})$.

CP for large-scale, sparse tensors has only recently been considered. Kolda et al. [136] developed a "greedy" CP for sparse tensors that computes one triad (i.e., rank-one component) at a time via an ALS method. In subsequent work, Kolda and Bader [135, 17] adapted the standard ALS algorithm in Figure 3.3 to sparse tensors. Zhang and Golub [244] propose a generalized Rayleigh-Newton iteration to compute a rank-one factor, which is another way to compute greedy CP. Kofidis and Regalia [131] present a higher-order power method for supersymmetric tensors.

We conclude this section by noting that there have also been substantial developments on variations of CP to account for missing values (e.g., [222]), to enforce nonnegativity constraints (see, e.g., §5.6), and for other purposes.

3.5. Applications of CP. CP's origins began in psychometrics in 1970. Carroll and Chang [38] introduced CANDECOMP in the context of analyzing multiple similarity or dissimilarity matrices from a variety of subjects. The idea was that simply averaging the data for all the subjects annihilated different points of view. They applied the method to one data set on auditory tones from Bell Labs and to another data set of comparisons of countries. Harshman [90] introduced PARAFAC because it eliminated the ambiguity associated with two-dimensional PCA and thus has better uniqueness properties. He was motivated by Cattell's principle of Parallel Proportional Profiles [40]. He applied it to vowel-sound data where different individuals (mode 1) spoke different vowels (mode 2) and the formant (i.e., the pitch) was measured (mode 3).

Appellof and Davidson [13] pioneered the use of CP in chemometrics in 1981. Andersson and Bro [9] survey its use in chemometrics to date. In particular, CP has proved useful in the modeling of fluorescence excitation-emission data.

Sidiropoulos, Bro, and Giannakis [196] considered the application of CP to sensor array processing. Other applications in telecommunications include [198, 197, 59]. CP also has important applications in independent component analysis (ICA); see [58] and references therein.

Several authors have used CP decompositions in neuroscience. As mentioned previously, Möcks [166] independently discovered CP in the context of event-related potentials in brain imaging; this work also included results on uniqueness and remarks about how to choose the number of factors. Later work by Field and Graupe [79] put the work of Möcks in context with the other work on CP and discussed practical aspects of working with data (scaling, etc.) and gave examples illustrating the utility of the CP decomposition for event-related potentials. Anderson and Rayens [8] apply

Preprint of article to appear in SIAM Review (June 10, 2008).

CP to fMRI data arranged as voxels by time by run and also as voxels by time by trial by run. Martínez-Montes et al. [163, 165] apply CP to a time-varying EEG spectrum arranged as a three-dimensional array with modes corresponding to time, frequency, and channel. Mørup et al. [170] looked at a similar problem and, moreover, Mørup, Hansen, and Arnfred [169] have released a MATLAB toolbox called ERPWAVELAB for multi-channel analysis of time-frequency transformed event related activity of EEG and MEG data. Recently, Acar et al. [2] and De Vos et al. [70, 71] have used CP for analyzing epileptic seizures. Stegeman [203] explains the differences between a three-way extension of indenpendent component analysis (ICA) and CP for multi-subject fMRI data in terms of the higher-order statistical properties.

The first application of tensors in data mining was by Acar et al. [4, 5], who applied different tensor decompositions, including CP, to the problem of discussion detanglement in online chatrooms. In text analysis, Bader, Berry, and Browne [14] used CP for automatic conversation detection in email over time using a term-by-author-by-time array.

Shashua and Levin [194] applied CP to image compression and classification. Furukawa et al. [82] have applied a CP model to bi-directional texture functions in order to build a compressed texture database. Bauckhage [19] extends discriminant analysis to higher-order data (color images, in this case) for classification.

Beylkin and Mohlenkamp [22, 23] apply CP to operators and conjecture that the border rank (which they call *optimal separation rank*) is low for many common operators. They provide an algorithm for computing the rank and prove that the multiparticle Schrödinger operator and Inverse Laplacian operators have ranks proportional to the log of the dimension. CP approximations have proven useful in approximating certain multi-dimensional operators such as the Newton potential; see Hackbusch, Khoromskij, and Tyrtyshnikov [89] and Hackbusch and Khoromskij [88]. Very recent work has focused on applying CP decompositions to stochastic PDEs [242, 73].

4. Compression and the Tucker decomposition. The Tucker decomposition was first introduced by Tucker in 1963 [224] and refined in subsequent articles by Levin [153] and Tucker [225, 226]. Tucker's 1966 article [226] is the most comprehensive of the early literature and is generally the one most cited. Like CP, the Tucker decomposition goes by many names, some of which are summarized in Table 4.1.

Name	Proposed by
Three-mode factor analysis (3MFA/Tucker3)	Tucker, 1966 [226]
Three-mode principal component analysis (3MPCA)	Kroonenberg and De Leeuw, 1980 [140]
N-mode principal components analysis	Kapteyn et al., 1986 [113]
Higher-order SVD (HOSVD)	De Lathauwer et al., 2000 [63]
N-mode SVD	Vasilescu and Terzopoulos, 2002 [229]

Table 4.1: Names for the Tucker decomposition (some specific to three-way and some for N-way).

The Tucker decomposition is a form of higher-order principal component analysis. It decomposes a tensor into a core tensor multiplied (or transformed) by a matrix along each mode. Thus, in the three-way case where $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we have

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}\, \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r = [\![\mathcal{G} \,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \qquad (4.1)$$

Preprint of article to appear in SIAM Review (June 10, 2008).

Here, $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are the factor matrices (which are usually orthogonal) and can be thought of as the principal components in each mode. The tensor $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ is called the *core tensor* and its entries show the level of interaction between the different components. The last equality uses the shorthand $[\![\mathcal{G}\,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ introduced in Kolda [134].

Elementwise, the Tucker decomposition in (4.1) is

$$x_{ijk} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}\, a_{ip}\, b_{jq}\, c_{kr}, \quad \text{for} \quad i = 1, \ldots, I, j = 1, \ldots, J, k = 1, \ldots, K.$$

Here P, Q, and R are the number of components (i.e., columns) in the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. If P, Q, R are smaller than I, J, K, the core tensor $\mathcal{G}$ can be thought of as a compressed version of $\mathcal{X}$. In some cases, the storage for the decomposed version of the tensor can be significantly smaller than for the original tensor; see Bader and Kolda [17]. The Tucker decomposition is illustrated in Figure 4.1.

Most fitting algorithms (discussed in §4.2) assume that the factor matrices are columnwise orthonormal, but it is not required. In fact, CP can be viewed as a special case of Tucker where the core tensor is superdiagonal and $P = Q = R$.

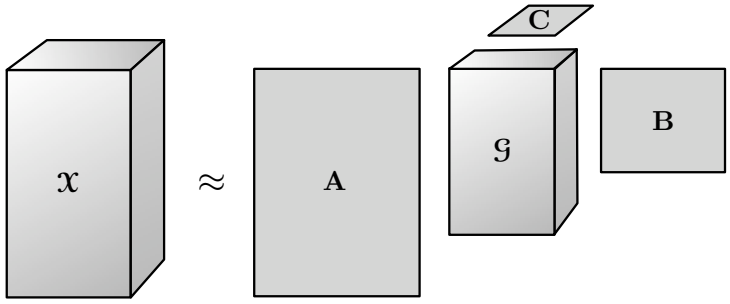


Fig. 4.1: Tucker decomposition of a three-way array

The matricized forms (one per mode) of (4.1) are

$$\mathbf{X}_{(1)} \approx \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^{\mathsf{T}},$$
$$\mathbf{X}_{(2)} \approx \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^{\mathsf{T}},$$
$$\mathbf{X}_{(3)} \approx \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^{\mathsf{T}}.$$

These equations follow from the the formulas in §2.4 and §2.6; see [134] for further details.

Though it was introduced in the context of three modes, the Tucker model can and has been generalized to N-way tensors [113] as:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} = [\![\mathcal{G}\,; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!], \tag{4.2}$$

or, elementwise, as

$$x_{i_1 i_2 \cdots i_N} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \cdots r_N}\, a^{(1)}_{i_1 r_1}\, a^{(2)}_{i_2 r_2} \cdots a^{(N)}_{i_N r_N}$$
$$\text{for} \quad i_n = 1, \ldots, I_n,\ n = 1, \ldots, N.$$

Preprint of article to appear in SIAM Review (June 10, 2008).

The matricized version of (4.2) is

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)}\mathbf{G}_{(n)}(\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^{\mathsf{T}}.$$

Two important variations of the decomposition are also worth noting here. The **Tucker2** decomposition [226] of a third-order array sets one of the factor matrices to be the identity matrix. For instance, a **Tucker2** decomposition is

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} = [\![\mathcal{G}\,; \mathbf{A}, \mathbf{B}, \mathbf{I}]\!].$$

This is the same as (4.1) except that $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ with $R = K$ and $\mathbf{C} = \mathbf{I}$, the $K \times K$ identity matrix. Likewise, the **Tucker1** decomposition [226] sets two of the factor matrices to be the identity matrix. For example, if the second and third factor matrices are the identity matrix, then we have

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} = [\![\mathcal{G}\,; \mathbf{A}, \mathbf{I}, \mathbf{I}]\!].$$

This is equivalent to standard two-dimensional PCA since

$$\mathbf{X}_{(1)} = \mathbf{A}\mathbf{G}_{(1)}.$$

These concepts extend easily to the N-way case — we can set any subset of the factor matrices to the identity matrix.

There are obviously many choices for tensor decompositions, which may lead to confusion about which model to choose for a particular application. Ceulemans and Kiers [42] discuss methods for choosing between CP and the different Tucker models in the three-way case.

4.1. The n-rank. Let $\mathcal{X}$ be an Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$. Then the *$n$-rank* of $\mathcal{X}$, denoted $\text{rank}_n(\mathcal{X})$, is the column rank of $\mathbf{X}_{(n)}$. In other words, the n-rank is the dimension of the vector space spanned by the mode-n fibers (see Figure 2.1). If we let $R_n = \text{rank}_n(\mathcal{X})$ for $n = 1, \ldots, N$, then we can say that $\mathcal{X}$ is a rank-$(R_1, R_2, \ldots, R_N)$ tensor, though n-mode rank should not be confused with the idea of rank (i.e., the minimum number of rank-1 components); see §3.1. Trivially, $R_n \leq I_n$ for all $n = 1, \ldots, N$.

Kruskal [143] introduced the idea of n-rank, and it was further popularized by De Lathauwer et al. [63]. The more general concept of *multiplex rank* was introduced much earlier by Hitchcock [106]. The difference is that n-mode rank uses only the mode-n unfolding of the tensor $\mathcal{X}$ whereas the multiplex rank can correspond to any arbitrary matricization (see §2.4).

For a given tensor $\mathcal{X}$, we can easily find an exact Tucker decomposition of rank-$(R_1, R_2, \ldots, R_N)$ where $R_n = \text{rank}_n(\mathcal{X})$. If, however, we compute a Tucker decomposition of rank-$(R_1, R_2, \ldots, R_N)$ where $R_n < \text{rank}_n(\mathcal{X})$ for one or more n, then it will be necessarily inexact and more difficult to compute. Figure 4.2 shows a *truncated* Tucker decomposition (not necessarily obtained by truncating an exact decomposition) which does not exactly reproduce $\mathcal{X}$.

4.2. Computing the Tucker decomposition. In 1966, Tucker [226] introduced three methods for computing a Tucker decomposition, but he was somewhat hampered by the computing ability of the day, stating that calculating the eigendecomposition for a 300×300 matrix "may exceed computer capacity." The first method in [226] is shown in Figure 4.3. The basic idea is to find those components that best capture the

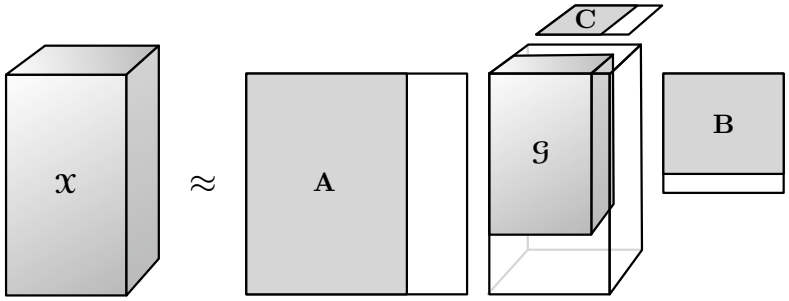


Fig. 4.2: Truncated Tucker decomposition of a three-way array

procedure HOSVD($\mathcal{X}$,R_1,R_2,...,R_N)
  for $n = 1, \ldots, N$ **do**
    $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{X}_{(n)}$
  end for
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
  return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Fig. 4.3: Tucker's "Method I" for computing a rank-$(R_1, R_2, \ldots, R_N)$ Tucker decomposition, later known as the HOSVD.

variation in mode n, independent of the other modes. Tucker presented it only for the three-way case, but the generalization to N ways is straightforward. This is sometimes referred to as the "Tucker1" method, though it is not clear whether this is because a Tucker1 factorization (discussed above) is computed for each mode or it was Tucker's first method. Today, this method is better known as the *higher-order SVD* (HOSVD) from the work of De Lathauwer, De Moor, and Vandewalle [63], who showed that the HOSVD is a convincing generalization of the matrix SVD and discussed ways to more efficiently compute the leading left singular vectors of $\mathbf{X}_{(n)}$. When $R_n < \text{rank}_n(\mathcal{X})$ for one or more n, the decomposition is called the *truncated* HOSVD. In fact, the core tensor of the HOSVD is *all-orthogonal*, which has relevance to truncating the decomposition; see [63] for details.

The truncated HOSVD is not optimal in terms of giving the best fit as measured by the norm of the difference, but it is a good starting point for an iterative alternating least squares (ALS) algorithm. In 1980, Kroonenberg and De Leeuw [140] developed an ALS algorithm, called TUCKALS3, for computing a Tucker decomposition for three-way arrays. (They also had a variant called TUCKALS2 that computed the Tucker2 decomposition of a three-way array.) Kapteyn, Neudecker, and Wansbeek [113] later extended TUCKALS3 to N-way arrays for $N > 3$. De Lathauwer, De Moor, and Vandewalle [64] proposed more efficient techniques for calculating the factor matrices (specifically, computing only the dominant singular vectors of $\mathbf{X}_{(n)}$ and using an SVD rather than an eigenvalue decomposition or even just computing an orthonormal basis of the dominant subspace) and called it the *Higher-order Orthogonal Iteration* (HOOI); see Figure 4.4. If we assume that $\mathcal{X}$ is a tensor of size

$I_1 \times I_2 \times \cdots \times I_N$, then the optimization problem that we wish to solve is

$$\begin{aligned} \min_{\mathcal{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} \quad & \left\| \mathcal{X} - [\![\mathcal{G} ; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \right\| \\ \text{subject to} \quad & \mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}, \\ & \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ and columnwise orthogonal for } n = 1, \ldots, N. \end{aligned} \tag{4.3}$$

By rewriting the above objective function in vectorized form as

$$\left\| \operatorname{vec}(\mathcal{X}) - (\mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}) \operatorname{vec}(\mathcal{G}) \right\|,$$

it is straightforward to show that the core tensor $\mathcal{G}$ must satisfy

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}.$$

We can then rewrite the (square of the) objective function as:

$$\begin{aligned} & \left\| \mathcal{X} - [\![\mathcal{G} ; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \right\|^2 \\ & \quad = \|\mathcal{X}\|^2 - 2\langle \mathcal{X}, [\![\mathcal{G} ; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \rangle + \|[\![\mathcal{G} ; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]\|^2 \\ & \quad = \|\mathcal{X}\|^2 - 2\langle \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}, \mathcal{G} \rangle + \|\mathcal{G}\|^2 \\ & \quad = \|\mathcal{X}\|^2 - 2\langle \mathcal{G}, \mathcal{G} \rangle + \|\mathcal{G}\|^2 \\ & \quad = \|\mathcal{X}\|^2 - \|\mathcal{G}\|^2 \\ & \quad = \|\mathcal{X}\|^2 - \|\mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}\|^2. \end{aligned}$$

The details of this transformation are readily available; see, e.g., [10, 64, 134]. Once again, we can use an alternating least squares approach to solve (4.3). Because $\|\mathcal{X}\|^2$ is constant, (4.3) can be recast as a series of subproblems involving the following maximization problem, which solves for the nth component matrix:

$$\begin{aligned} \max_{\mathbf{A}^{(n)}} \quad & \left\| \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}} \right\| \\ \text{subject to} \quad & \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ and columnwise orthogonal.} \end{aligned} \tag{4.4}$$

The objective function in (4.4) can be rewritten in matrix form as

$$\left\| \mathbf{A}^{(n)\mathsf{T}} \mathbf{W} \right\| \text{ with } \mathbf{W} = \mathbf{X}_{(n)} (\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}).$$

The solution can be determined using the SVD; simply set $\mathbf{A}^{(n)}$ to be the R_n leading left singular vectors of $\mathbf{W}$. This method will converge to a solution where the objective function of (4.3) ceases to decrease, but it is not guaranteed to converge to the global optimum or even a stationary point of (4.3) [140, 64]. Andersson and Bro [10] consider methods for speeding up the HOOI algorithm such as how to do the computations, how to initialize the method, and how to compute the singular vectors.

Recently, Eldén and Savas [77] proposed a Newton-Grassmann optimization approach for computing a Tucker decomposition of a 3-way tensor. The problem is cast as a nonlinear program with the factor matrices constrained to a Grassmann manifold that defines an equivalence class of matrices with orthonormal columns. The

Preprint of article to appear in SIAM Review (June 10, 2008).

procedure HOOI($\mathcal{X}$,R_1,R_2,...,R_N)
  initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$ using HOSVD
  repeat
    for $n = 1, \ldots, N$ **do**
      $\mathcal{Y} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \cdots \times_{n-1} \mathbf{A}^{(n-1)\mathsf{T}} \times_{n+1} \mathbf{A}^{(n+1)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
      $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{Y}_{(n)}$
    end for
  until fit ceases to improve or maximum iterations exhausted
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
  return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Fig. 4.4: Alternating least squares algorithm to compute a rank-$(R_1, R_2, \ldots, R_N)$ Tucker decomposition for an Nth order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$. Also known as the higher-order orthogonal iteration.

Newton-Grassmann approach takes many fewer iterations than HOOI and demonstrates quadratic convergence numerically, though each iteration is more expensive than HOOI due to the computation of the Hessian. This method will converge to a stationary point of (4.3).

The question of how to choose the rank has been addressed by Kiers and Der Kinderen [123] who have a straightforward procedure (cf., CONCORDIA for CP in §3.4) for choosing the appropriate rank of a Tucker model based on an HOSVD calculation.

4.3. Lack of uniqueness and methods to overcome it. Tucker decompositions are not unique. Consider the three-way decomposition in (4.1). Let $\mathbf{U} \in \mathbb{R}^{P \times P}$, $\mathbf{V} \in \mathbb{R}^{Q \times Q}$, and $\mathbf{W} \in \mathbb{R}^{R \times R}$ be nonsingular matrices. Then

$$[\![\mathcal{G}\,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = [\![\mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\,; \mathbf{A}\mathbf{U}^{-1}, \mathbf{B}\mathbf{V}^{-1}, \mathbf{C}\mathbf{W}^{-1}]\!].$$

In other words, we can modify the core $\mathcal{G}$ without affecting the fit so long as we apply the inverse modification to the factor matrices.

This freedom opens the door to choosing transformations that *simplify* the core structure in some way so that most of the elements of $\mathcal{G}$ are zero, thereby eliminating interactions between corresponding components and improving uniqueness. Superdiagonalization of the core is impossible (even in the symmetric case; see [50, 49]), but it is possible to try to make as many elements zero or very small as possible. This was first observed by Tucker [226] and has been studied by several authors; see, e.g., [117, 103, 127, 172, 12]. One possibility is to apply a set of orthogonal rotations that optimizes a function that measures the "simplicity" of the core as measured by some objective [120]. Another is to use a Jacobi-type algorithm to maximize the magnitude of the diagonal entries [46, 65, 162]. Finally, the HOSVD generates an all-orthogonal core, as mentioned previously, which is yet another type of special core structure that might be useful.

4.4. Applications of Tucker. Several examples of using the Tucker decomposition in chemical analysis are provided by Henrion [104] as part of a tutorial on N-way PCA. Examples from psychometrics are provided by Kiers and Van Mechelen [129] in their overview of three-way component analysis techniques. The overview is a good introduction to three-way methods, explaining when to use three-way techniques rather than two-way (based on an ANOVA test), how to preprocess the data, guidance on choosing the rank of the decomposition and an appropriate rotation, and methods for presenting the results.

Preprint of article to appear in SIAM Review (June 10, 2008).

De Lathauwer and Vandewalle [68] consider applications of the Tucker decomposition to signal processing. Muti and Bourennane [173] have applied the Tucker decomposition to extend Wiener filters in signal processing.

Vasilescu and Terzopoulos [229] pioneered the use of Tucker decompositions in computer vision with TensorFaces. They considered facial image data from multiple subjects where each subject had multiple pictures taken under varying conditions. For instance, if the variation is the lighting, the data would be arranged into three modes: person, lighting conditions, and pixels. Additional modes such as expression, camera angle, and more can also be incorporated. Recognition using TensorFaces is significantly more accurate than standard PCA techniques [230]. TensorFaces is also useful for compression and can remove irrelevant effects, such as lighting, while retaining key facial features [231]. Vasilescu [228] has also applied the Tucker decomposition to human motion. Wang and Ahuja [236, 237] used Tucker to model facial expressions and for image data compression, and Vlassic et al. [235] use Tucker to transfer facial expressions. Nagy and Kilmer [174] use the Tucker decomposition to construct Kronecker product approximations for preconditioners in image processing. Vasilescu and Terzopolous [232] have also applied the Tucker decomposition to the bidirectional texture function (BTF) for rendering texture in two-dimensional images. Many related applications exist, such as watermarking MPEG videos using Tucker [1].

Grigorascu and Regalia [87] consider extending ideas from structured matrices to tensors. They are motivated by the structure in higher-order cumulants and corresponding polyspectra in applications and develop an extension of a Schur-type algorithm. Langville and Stewart [149] develop preconditioners for stochastic automata networks (SANs) based on Tucker decompositions of the matrices rearranged as tensors. Khoromskij and Khoromskaia [115] have applied the Tucker decomposition to approximations of classical potentials including optimized algorithms.

In data mining, Savas and Eldén [190, 191] applied the HOSVD to the problem of identifying handwritten digits. As mentioned previously, Acar et al. [4, 5], applied different tensor decompositions, including Tucker, to the problem of discussion detanglement in online chatrooms. J.-T. Sun et al. [211] used Tucker to analyze web site click-through data. Liu et al. [157] applied Tucker to create a tensor space model, analogous to the well-known vector space model in text analysis. J. Sun et al. [209, 210] have written a pair of papers on dynamically updating a Tucker approximation, with applications ranging from text analysis to environmental and network modeling.

5. Other decompositions. There are a number of other tensor decompositions related to CP and Tucker. Most of these decompositions originated in the psychometrics and applied statistics communities and have only recently become more widely known in other fields such as chemometrics and social network analysis.

We list the decompositions discussed in this section in Table 5.1. We describe the decompositions in approximately chronological order; for each decomposition, we survey its origins, briefly discuss computation, and describe some applications. We finish this section with a discussion of nonnegative tensor decompositions and a few more decompositions that are not covered in depth in this review.

5.1. INDSCAL. Individual Differences in Scaling (INDSCAL) is a special case of CP for three-way tensors that are symmetric in two modes; see §2.2. It was proposed by Carroll and Chang [38] in the same paper in which they introduced CANDECOMP; see §3.

For INDSCAL, the first two factor matrices in the decomposition are constrained

Preprint of article to appear in SIAM Review (June 10, 2008).

Name	Proposed by
Individual Differences in Scaling (INDSCAL)	Carroll and Chang, 1970 [38]
Parallel Factors for Cross Products (PARAFAC2)	Harshman, 1972 [92]
CANDECOMP with Linear Constraints (CANDELINC)	Carroll et al., 1980 [39]
Decomposition into Directional Components (DEDICOM)	Harshman, 1978 [93]
PARAFAC and Tucker2 (PARATUCK2)	Harshman and Lundy, 1996 [100]

Table 5.1: Other tensor decompositions.

to be equal, at least in the final solution. Without loss of generality, we assume that the first two modes are symmetric. Thus, for a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ with $x_{ijk} = x_{jik}$ for all i, j, k, the INDSCAL model is given by

$$\mathcal{X} \approx [\![\mathbf{A}, \mathbf{A}, \mathbf{C}]\!] = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{a}_r \circ \mathbf{c}_r. \tag{5.1}$$

Applications involving symmetric slices are common, especially when dealing with similarity, dissimilarity, distance, or covariance matrices.

INDSCAL is generally computed using a procedure to compute CP. The two $\mathbf{A}$ matrices are treated as distinct factors ($\mathbf{A}_L$ and $\mathbf{A}_R$, for left and right, respectively) and updated separately, without an explicit constraint enforcing equality. Though the estimates early in the process are different, the hope is that the inherent symmetry of the data causes the two factors to eventually converge to be equal, up to scaling by a diagonal matrix. In other words,

$$\mathbf{A}_L = \mathbf{D}\mathbf{A}_R,$$
$$\mathbf{A}_R = \mathbf{D}^{-1}\mathbf{A}_L,$$

where $\mathbf{D}$ is an $R \times R$ diagonal matrix. In practice, the last step is to set $\mathbf{A}_L = \mathbf{A}_R$ (or vice versa) and calculate $\mathbf{C}$ one last time [38]. In fact, Ten Berge et al. [217] have shown that equality does not always happen in practice. The best method for computing INDSCAL is still an open question.

Ten Berge, Sidiropoulos, and Rocci [219] have studied the typical and maximal rank of partially symmetric three-way tensors of size $I \times 2 \times 2$ and $I \times 3 \times 3$ (see Table 3.5 in §3.1). The typical rank of these tensors is the same or smaller than the typical rank of nonsymmetric tensors of the same size. Moreover, the CP solution yields the INDSCAL solution whenever the CP solution is unique and, surprisingly, still does quite well even in the case of non-uniqueness. These results are restricted to very specific cases, and Ten Berge et al. [219] note that much more general results are needed. INDSCAL uniqueness is further studied in [207]. Stegeman [204] studies degeneracy for certain cases of INDSCAL.

5.2. PARAFAC2. PARAFAC2 [92] is not strictly a tensor decomposition. Rather, it is a variant of CP that can be applied to a collection of matrices that each have the same number of columns but a different number of rows. Here we apply PARAFAC2 to a set of matrices $\mathbf{X}_k$, for $k = 1, \ldots, K$, such that each $\mathbf{X}_k$ is of size $I_k \times J$, where I_k is allowed to vary with k.

Essentially, PARAFAC2 relaxes some of CP's constraints. Whereas CP applies the same factors across a parallel set of matrices, PARAFAC2 instead applies the same factor along one mode and allows the other factor matrix to vary. An advantage

Preprint of article to appear in SIAM Review (June 10, 2008).

of PARAFAC2 is that, not only can it approximate data in a regular three-way tensor with fewer constraints than CP, it can also be applied to a collection of matrices with varying sizes in one mode, e.g., same column dimension but different row size.

Let R be the number of dimensions of the decomposition. Then the PARAFAC2 model has the following form:

$$\mathbf{X}_k \approx \mathbf{U}_k \mathbf{S}_k \mathbf{V}^\mathsf{T}, \qquad k = 1, \ldots, K. \tag{5.2}$$

where $\mathbf{U}_k$ is an $I_k \times R$ matrix and $\mathbf{S}_k$ is a $R \times R$ diagonal matrix for $k = 1, \ldots, K$, and $\mathbf{V}$ is a $J \times R$ factor matrix that does not vary with k. The general PARAFAC2 model for a collection of matrices with varying sizes is shown in Figure 5.1.

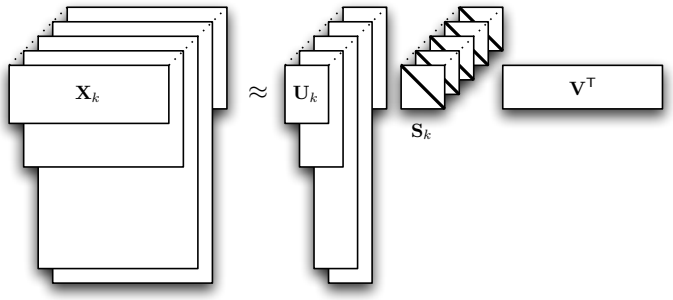


Fig. 5.1: Illustration of PARAFAC2.

PARAFAC2 is not unique without additional constraints. For example, if $\mathbf{T}$ is an $R \times R$ nonsingular matrix and $\mathbf{F}_k$ is an $R \times R$ diagonal matrix for $k = 1, \ldots, K$, then

$$\mathbf{U}_k \mathbf{S}_k \mathbf{V}^\mathsf{T} = (\mathbf{U}_k \mathbf{S}_k \mathbf{T}^{-1} \mathbf{F}_k^{-1}) \mathbf{F}_k (\mathbf{V}\mathbf{T}^\mathsf{T})^\mathsf{T} = \mathbf{G}_k \mathbf{F}_k \mathbf{W}^\mathsf{T}$$

is an equally valid decomposition. Consequently, to improve the uniqueness properties, Harshman [92] imposed the constraint that the cross product $\mathbf{U}_k^\mathsf{T}\mathbf{U}_k$ is constant over k, i.e., $\mathbf{\Phi} = \mathbf{U}_k^\mathsf{T}\mathbf{U}_k$ for $k = 1, \ldots, K$. Thus, with this constraint, PARAFAC2 can be expressed as

$$\mathbf{X}_k \approx \mathbf{Q}_k \mathbf{H} \mathbf{S}_k \mathbf{V}^\mathsf{T}, \qquad k = 1, \ldots, K. \tag{5.3}$$

Here, $\mathbf{U}_k = \mathbf{Q}_k\mathbf{H}$ where $\mathbf{Q}_k$ is of size $I_k \times R$ and constrained to be orthonormal, and $\mathbf{H}$ is an $R \times R$ matrix that does not vary by slice. The cross-product constraint is enforced implicitly since

$$\mathbf{U}_k^\mathsf{T}\mathbf{U}_k = \mathbf{H}^\mathsf{T}\mathbf{Q}_k^\mathsf{T}\mathbf{Q}_k\mathbf{H} = \mathbf{H}^\mathsf{T}\mathbf{H} = \mathbf{\Phi}.$$

5.2.1. Computing PARAFAC2. Algorithms for fitting PARAFAC2 either fit the cross-products of the covariance matrices [92, 118] (*indirect fitting*) or fit (5.3) to the original data itself [126] (*direct fitting*). The indirect fitting approach finds $\mathbf{V}$, $\mathbf{S}_k$ and $\mathbf{\Phi}$ corresponding to the cross products:

$$\mathbf{X}_k^\mathsf{T}\mathbf{X}_k \approx \mathbf{V}\mathbf{S}_k\mathbf{\Phi}\mathbf{S}_k\mathbf{V}^\mathsf{T}, \qquad k = 1, \ldots, K.$$

This can be done by using a DEDICOM decomposition (see §5.4) with positive semi-definite constraints on $\mathbf{\Phi}$. The direct fitting approach solves for the unknowns in a

two-step iterative approach, first by finding $\mathbf{Q}_k$ from a minimization using the SVD and then updating the remaining unknowns, $\mathbf{H}$, $\mathbf{S}_k$, and $\mathbf{V}$, using one step of a CP-ALS procedure. See [126] for details.

PARAFAC2 is (essentially) unique under certain conditions pertaining to the number of matrices (K), the positive definiteness of $\mathbf{\Phi}$, full column rank of $\mathbf{A}$, and nonsingularity of $\mathbf{S}_k$ [100, 215, 126].

5.2.2. PARAFAC2 applications. Bro et al. [31] use PARAFAC2 to handle time shifts in resolving chromatographic data with spectral detection. In this application, the first mode corresponds to elution time, the second mode to wavelength, and the third mode to samples. The PARAFAC2 model does not assume parallel proportional elution profiles but rather that the matrix of elution profiles preserves its "inner-product structure" across samples, which means that the cross-product of the corresponding factor matrices in PARAFAC2 is constant across samples.

Wise, Gallager, and Martin [240] applied PARAFAC2 to the problem of fault detection in a semi-conductor etch process. In this application (like many others), there is a difference in the number of time steps per batch. Alternative methods such as time warping alter the original data. The advantage of PARAFAC2 is that it can be used on the original data.

Chew et al. [44] use PARAFAC2 for clustering documents across multiple languages. The idea is to extend the concept of latent semantic indexing [75, 72, 21] in a cross-language context by using multiple translations of the same collection of documents, i.e., a parallel corpus. In this case, $\mathbf{X}_k$ is a term-by-document matrix for the kth language in the parallel corpus. The number of terms varies considerably by language. The resulting PARAFAC2 decomposition is such that the document-concept matrix $\mathbf{V}$ is the same across all languages, while each language has its own term-concept matrix $\mathbf{Q}_k$.

5.3. CANDELINC. A principal problem in multidimensional analysis is the interpretation of the factor matrices from tensor decompositions. Consequently, including domain or user knowledge is desirable, and this can be done by imposing constraints. CANDELINC (canonical decomposition with linear constraints) is CP with linear constraints on one or more of the factor matrices and was introduced by Carroll, Pruzansky, and Kruskal [39]. Though a constrained model may not explain as much variance in the data (i.e., it may have a larger residual error), the decomposition is often more meaningful and interpretable.

For instance, in the three-way case, CANDELINC requires that the CP factor matrices from (3.4) satisfy

$$\mathbf{A} = \mathbf{\Phi_A}\hat{\mathbf{A}}, \quad \mathbf{B} = \mathbf{\Phi_B}\hat{\mathbf{B}}, \quad \mathbf{C} = \mathbf{\Phi_C}\hat{\mathbf{C}}. \tag{5.4}$$

Here, $\mathbf{\Phi_A} \in \mathbb{R}^{I\times M}$, $\mathbf{\Phi_B} \in \mathbb{R}^{J\times N}$, and $\mathbf{\Phi_C} \in \mathbb{R}^{K\times P}$ define the column space for each factor matrix, and $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$ are the constrained solutions as defined below. Thus, the CANDELINC model, which is (3.4) coupled with (5.4), is given by

$$\mathcal{X} \approx [\![\mathbf{\Phi_A}\hat{\mathbf{A}}, \mathbf{\Phi_B}\hat{\mathbf{B}}, \mathbf{\Phi_C}\hat{\mathbf{C}}]\!]. \tag{5.5}$$

Without loss of generality, the constraint matrices $\mathbf{\Phi_A}, \mathbf{\Phi_B}, \mathbf{\Phi_C}$ are assumed to be orthonormal. We can make this assumption because any matrix that does not satisfy the requirement can be replaced by one that generates the same column space and is columnwise orthogonal.

Preprint of article to appear in SIAM Review (June 10, 2008).

5.3.1. Computing CANDELINC. Fitting CANDELINC is straightforward. Under the assumption that the constraint matrices are orthonormal, we can simply compute CP on the projected tensor. For example, in the third-order case discussed above, the projected $M \times N \times P$ tensor is given by

$$\hat{\mathcal{X}} = \mathcal{X} \times_1 \mathbf{\Phi}_{\mathbf{A}}^{\mathsf{T}} \times_2 \mathbf{\Phi}_{\mathbf{B}}^{\mathsf{T}} \times_3 \mathbf{\Phi}_{\mathbf{C}}^{\mathsf{T}} = [\![\mathcal{X}\,; \mathbf{\Phi}_{\mathbf{A}}^{\mathsf{T}}, \mathbf{\Phi}_{\mathbf{B}}^{\mathsf{T}}, \mathbf{\Phi}_{\mathbf{C}}^{\mathsf{T}}]\!].$$

We then compute its CP decomposition to get:

$$\hat{\mathcal{X}} \approx [\![\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}]\!]. \tag{5.6}$$

The solution to the original problem is obtained by multiplying each factor matrix of the projected tensor by the corresponding constraint matrix as in (5.5).

5.3.2. CANDELINC and compression. If $M \ll I$, $N \ll J$, and $P \ll K$, then the projected tensor $\hat{\mathcal{X}}$ is much smaller than the original tensor $\mathcal{X}$, and so decompositions of $\hat{\mathcal{X}}$ are typically much faster. Indeed, CANDELINC is the theoretical base for a procedure that computes CP as follows [124, 30]. First, a Tucker decomposition is applied to a given data tensor in order to compress it to size $M \times N \times P$. Second, the CP decomposition of the resulting "small" core tensor is computed. Third, this is translated to a CP decomposition of the original data tensor as described above, and then the final result may be refined by a few CP-ALS iterations on the full tensor. The utility of compression varies by application; Tomasi and Bro [223] report that compression does not necessarily lead to faster computational times because more ALS iterations are required on the compressed tensor.

5.3.3. CANDELINC applications. As mentioned above, the primary application of CANDELINC is computing CP for large-scale data via the Tucker compression technique mentioned above [124, 30]. Kiers developed a procedure involving compression and regularization to handle multicollinearity in chemometrics data [121]. Ibraghimov applies CANDELINC-like ideas to develop preconditioners for 3D integral operators [108]. Khoromskij and Khoromskaia [115] have applied CANDELINC (which they call "Tucker-to-canonical") to approximations of classical potentials.

5.4. DEDICOM. DEDICOM (decomposition into directional components) is a family of decompositions introduced by Harshman [93]. The idea is as follows. Suppose that we have I objects and a matrix $\mathbf{X} \in \mathbb{R}^{I \times I}$ that describes the *asymmetric* relationships between them. For instance, the objects might be countries and x_{ij} represents the value of exports from country i to country j. Typical factor analysis techniques either do not account for the fact that the two modes of a matrix may correspond to the same entities or that there may be directed interactions between them. DEDICOM, on the other hand, attempts to group the I objects into R latent components (or groups) and describe their pattern of interactions by computing $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{R} \in \mathbb{R}^{R \times R}$ such that

$$\mathbf{X} \approx \mathbf{A}\mathbf{R}\mathbf{A}^{\mathsf{T}}. \tag{5.7}$$

Each column in $\mathbf{A}$ corresponds to a latent component such that a_{ir} indicates the participation of object i in group r. The matrix $\mathbf{R}$ indicates the interaction between the different components, e.g., r_{ij} represents the exports from group i to group j.

There are two indeterminacies of scale and rotation that need to be addressed [93]. First, the columns of $\mathbf{A}$ may be scaled in a number of ways without affecting the solution. One choice is to have unit length in the 2-norm; other choices give rise

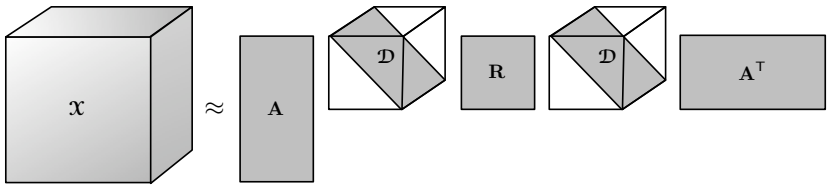


Fig. 5.2: Three-way DEDICOM model.

to different benefits of interpreting the results [95]. Second, the matrix $\mathbf{A}$ can be transformed with any nonsingular matrix $\mathbf{T}$ with no loss of fit to the data because $\mathbf{ARA}^\mathsf{T} = (\mathbf{AT})(\mathbf{T}^{-1}\mathbf{RT}^{-\mathsf{T}})(\mathbf{AT})^\mathsf{T}$. Thus, the solution obtained in $\mathbf{A}$ is not unique [95]. Nevertheless, it is standard practice to apply some accepted rotation to "fix" $\mathbf{A}$. A common choice is to adopt VARIMAX rotation [112] such that the variance across columns of $\mathbf{A}$ is maximized.

A further practice in some problems is to ignore the diagonal entries of $\mathbf{X}$ in the residual calculation [93]. For many cases, this makes sense because one wishes to ignore self-loops (e.g., a country does not export to itself). This is commonly handled by estimating the diagonal values from the current approximation $\mathbf{ARA}^\mathsf{T}$ and including them in $\mathbf{X}$.

Three-way DEDICOM [93] is a higher-order extension of the DEDICOM model that incorporates a third mode of the data. As with CP, adding a third dimension gives this decomposition stronger uniqueness properties [100]. Here we assume $\mathcal{X} \in \mathbb{R}^{I\times I\times K}$. In our previous example of trade among nations, the third mode may correspond to time. For instance, $k = 1$ corresponds to trade in 1995, $k = 2$ to 1996, and so on. The decomposition is then

$$\mathbf{X}_k \approx \mathbf{AD}_k\mathbf{RD}_k\mathbf{A}^\mathsf{T} \quad \text{for } k = 1,\ldots,K. \tag{5.8}$$

Here $\mathbf{A}$ and $\mathbf{R}$ are as in (5.7), except that $\mathbf{A}$ is not necessarily orthogonal. The matrices $\mathbf{D}_k \in \mathbb{R}^{R\times R}$ are diagonal, and entry $(\mathbf{D}_k)_{rr}$ indicates the participation of the rth latent component at time k. We can assemble the matrices $\mathbf{D}_k$ into a tensor $\mathcal{D} \in \mathbb{R}^{R\times R\times K}$. Unfortunately, we are constrained to slab notation (i.e., slice-by-slice) for expressing the model because DEDICOM cannot be expressed easily using more general notation. Three-way DEDICOM is illustrated in Figure 5.2.

For many applications, it is reasonable to impose nonnegativity constraints on $\mathcal{D}$ [92]. Dual-domain DEDICOM is a variation where the scaling array $\mathcal{D}$ and/or matrix $\mathbf{A}$ may be different on the left and right of $\mathbf{R}$. This form is encapsulated by PARATUCK2 (see §5.5).

5.4.1. Computing three-way DEDICOM. There are a number of algorithms for computing the two-way DEDICOM model, e.g., [128], and for variations such as constrained DEDICOM [125, 187]. For three-way DEDICOM, see Kiers [116, 118] and Bader, Harshman, and Kolda [15]. Because $\mathbf{A}$ and $\mathcal{D}$ appear on both the left and right, fitting three-way DEDICOM is a difficult nonlinear optimization problem with many local minima.

Kiers [118] presents an alternating least squares (ALS) algorithm that is efficient on small tensors. Each column of $\mathbf{A}$ is updated with its own least-squares solution while holding the others fixed. Each subproblem to compute one column of $\mathbf{A}$ involves a full eigendecomposition of a dense $I \times I$ matrix, which makes this procedure

Preprint of article to appear in SIAM Review (June 10, 2008).

expensive for large, sparse $\mathcal{X}$. In a similar alternating fashion, the elements of $\mathcal{D}$ are updated one at a time by minimizing a fourth degree polynomial. The best $\mathbf{R}$ for a given $\mathbf{A}$ and $\mathcal{D}$ is found from a least-squares solution using the pseudo-inverse of an $I^2 \times R^2$ matrix, which can be simplified to the inverse of an $R^2 \times R^2$ matrix.

Bader et al. [15] have proposed an algorithm called Alternating Simultaneous Approximation, Least Squares, and Newton (ASALSAN). The approach relies on the same update for $\mathbf{R}$ as in [118] but uses different methods for updating $\mathbf{A}$ and $\mathcal{D}$. Instead of solving for $\mathbf{A}$ column-wise, ASALSAN solves for all columns of $\mathbf{A}$ simultaneously using an approximate least-squares solution. Because there are RK elements of $\mathcal{D}$, which is not likely to be many, Newton's method is used to find all elements of $\mathcal{D}$ simultaneously. The same paper [15] introduces a nonnegative variant.

5.4.2. DEDICOM applications. Most of the applications of DEDICOM in the literature have focused on two-way (matrix) data, but there are some three-way applications. Harshman and Lundy [99] analyzed asymmetric measures of yearly trade (import-export) among a set of nations over a period of 10 years. Lundy et al. [160] presented an application of three-way DEDICOM to skew-symmetric data for paired preference ratings of treatments for chronic back pain, though they note that they needed to impose some constraints to get meaningful results.

Bader et al. [15] recently applied their ASALSAN method for computing DEDICOM on email communication graphs over time. In this case, x_{ijk} corresponded to the (scaled) number of email messages sent from person i to person j in month k.

5.5. PARATUCK2. Harshman and Lundy [100] introduced PARATUCK2, a generalization of DEDICOM that considers interactions between two possibly *different* sets of objects. The name is derived from the fact that this decomposition can be considered as a combination of CP and TUCKER2.

Given a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the goal is to group the mode-one objects into P latent components and the mode-two group into Q latent components. Thus, the PARATUCK2 decomposition is given by

$$\mathbf{X}_k \approx \mathbf{A}\mathbf{D}_k^A\mathbf{R}\mathbf{D}_k^B\mathbf{B}^\mathsf{T}, \quad \text{for } k = 1, \ldots, K. \tag{5.9}$$

Here, $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, $\mathbf{R} \in \mathbb{R}^{P \times Q}$, and $\mathbf{D}_k^A \in \mathbb{R}^{P \times P}$ and $\mathbf{D}_k^B \in \mathbb{R}^{Q \times Q}$ are diagonal matrices. As with DEDICOM, the columns of $\mathbf{A}$ and $\mathbf{B}$ correspond to the latent factors, so b_{jq} is the association of object j with latent component q. Likewise, the entries of the diagonal matrices $\mathbf{D}_k$ indicate the degree of participation for each latent component with respect to the third dimension. Finally, the rectangular matrix $\mathbf{R}$ represents the interaction between the P latent components in $\mathbf{A}$ and the Q latent components in $\mathbf{B}$. The matrices $\mathbf{D}_k^A$ and $\mathbf{D}_k^B$ can be stacked to form tensors $\mathcal{D}^A$ and $\mathcal{D}^B$ respectively. PARATUCK2 is illustrated in Figure 5.3.

Harshman and Lundy [100] prove the uniqueness of axis orientation for the general PARATUCK2 model and for the symmetrically weighted version (i.e., where $\mathcal{D}^A = \mathcal{D}^B$), all subject to the conditions that $P = Q$ and $\mathbf{R}$ has no zeros.

5.5.1. Computing PARATUCK2. Bro [28] discusses an alternating least-squares algorithm for computing the parameters of PARATUCK2. The algorithm follows in the same spirit as Kiers' DEDICOM algorithm [118] except that the problem is only linear in the parameters $\mathbf{A}, \mathcal{D}^A, \mathbf{B}, \mathcal{D}^B$, and $\mathbf{R}$ and involves only linear least-squares subproblems.

5.5.2. PARATUCK2 applications. Harshman and Lundy [100] proposed PARATUCK2 as a general model for a number of existing models, including PARAFAC2 (see §5.2)

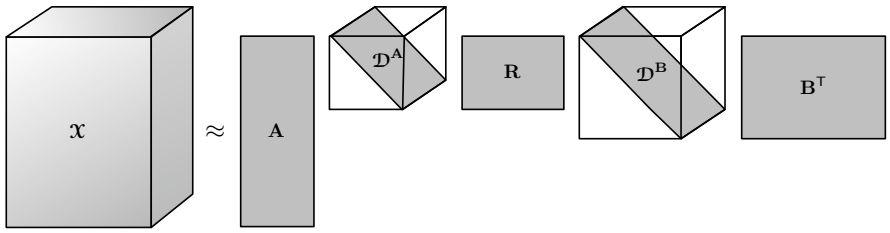


Fig. 5.3: General PARATUCK2 model.

and three-way DEDICOM (see §5.4). Consequently, there are few published applications of PARATUCK2 proper.

Bro [28] mentions that PARATUCK2 is appropriate for problems that involve interactions between factors or when more flexibility than CP is needed but not as much as Tucker. This can occur, for example, with rank-deficient data in physical chemistry, for which a slightly restricted form of PARATUCK2 is suitable. Bro [28] cites an example of the spectrofluorometric analysis of three types of cow's milk. Two of the three components are almost collinear, which means that CP would be unstable and could not identify the three components. A rank-$(2, 3, 3)$ analysis with a Tucker model is possible but would introduce rotational indeterminacy. A restricted form of PARATUCK2, with two columns in $\mathbf{A}$ and three in $\mathbf{B}$, is appropriate in this case.

5.6. Nonnegative tensor factorizations. Paatero and Tapper [181] and Lee and Seung [151] proposed using nonnegative matrix factorizations for analyzing nonnegative data, such as environmental models and grayscale images, because it is desirable for the decompositions to retain the nonnegative characteristics of the original data and thereby facilitate easier interpretation. It is also natural to impose nonnegativity constraints on tensor factorizations. Many papers refer to nonnegative tensor factorization generically as NTF but fail to differentiate between CP and Tucker. Hereafter, we use the terminology NNCP (nonnegative CP) and NNT (nonnegative Tucker).

Bro and De Jong [32] consider NNCP and NNT. They solve the subproblems in CP-ALS and Tucker-ALS with a specially adapted version of the NNLS method of Lawson and Hanson [150]. In the case of NNCP for a third-order tensor, subproblem (3.7) requires a least squares solution for $\mathbf{A}$. The change here is to impose nonnegativity constraints on $\mathbf{A}$.

Similarly, for NNCP, Friedlander and Hatz [81] solve a bound constrained linear least-squares problem. Additionally, they impose sparseness constraints by regularizing the nonnegative tensor factorization with an l_1-norm penalty function. While this function is nondifferentiable, it has the effect of "pushing small values exactly to zero while leaving large (and significant) entries relatively undisturbed." While the solution of the standard problem is unbounded due to the indeterminacy of scale, regularizing the problem has the added benefit of keeping the solution bounded. They demonstrate the effectiveness of their approach on image data.

Paatero [178] uses a Gauss-Newton method with a logarithmic penalty function to enforce the nonnegativity constraints in NNCP. The implementation is described in [179].

Welling and Weber [238] do multiplicative updates like Lee and Seung [151] for

Preprint of article to appear in SIAM Review (June 10, 2008).

NNCP. For instance, the update for $\mathbf{A}$ in third-order NNCP is given by:

$$a_{ir} \leftarrow a_{ir} \frac{(\mathbf{X}_{(1)}\mathbf{Z})_{ir}}{(\mathbf{A}\mathbf{Z}^{\mathsf{T}}\mathbf{Z})_{ir}}, \quad \text{where} \quad \mathbf{Z} = (\mathbf{C} \odot \mathbf{B}).$$

It is helpful to add a small number like $\epsilon = 10^{-9}$ to the denominator to add stability to the calculation and guard against introducing a negative number from numerical underflow. Welling and Weber's application is to image decompositions. FitzGerald, Cranitch, and Coyle [80] use a multiplicative update form of NNCP for sound source separation. Bader, Berry, and Browne [14] apply NNCP using multiplicative updates to discussion tracking in Enron email data. Mørup, Hansen, Parnas, and Arnfred [167] also develop a multiplicative update for NNCP. They consider both least squares and Kulbach-Leibler (KL) divergence, and the methods are applied to EEG data.

Shashua and Hazan [193] derive an EM-based method to calculate a rank-one NNCP decomposition. A rank-R NNCP model is calculated by doing a series of rank-one approximations to the residual. They apply the method to problems in computer vision. Hazan, Polak, and Shashua [102] apply the method to image data and observe that treating a set of images as a third-order tensor is better in terms of handling spatial redundancy in images, as opposed to using nonnegative matrix factorizations on a matrix of vectorized images. Shashua, Zass, and Hazan [195] also consider clustering based on nonnegative factorizations of supersymmetric tensors.

Mørup, Hansen, and Arnfred [168] develop multiplicative updates for a NNT decomposition and impose sparseness constraints to enhance uniqueness. Moreover, they observe that structure constraints can be imposed on the core tensor (to represent known interactions) and the factor matrices (per desired structure). They apply NNT with sparseness constraints to EEG data. Cichocki et al. [45] propose a nonnegative version of PARAFAC2 for EEG data. They develop a multiplicative update based on α- and β-divergences as well as a regularized ALS algorithm and an alternating interior-point gradient algorithm based on β-divergences.

Bader, Harshman, and Kolda [15] imposed nonnegativity constraints on the DEDICOM model using a nonnegative version of the ASALSAN method. They replaced the least squares updates with the multiplicative update introduced in [151]. The Newton procedure for updating $\mathcal{D}$ already employed nonnegativity constraints.

5.7. More decompositions. Recently, several different groups of researchers have proposed models that combine aspects of CP and Tucker. Recall that CP expresses a tensor as the sum of rank-one tensors. In these newer models, the tensor is expressed as a sum of low-rank Tucker tensors. In other words, for a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we have

$$\mathcal{X} \approx \sum_{r=1}^{R} [\![\mathcal{G}_r \, ; \mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r]\!]. \tag{5.10}$$

Here we assume $\mathcal{G}_r$ is of size $M_r \times N_r \times P_r$, $\mathbf{A}_r$ is of size $I \times M_r$, $\mathbf{B}_r$ is of size $J \times N_r$, and $\mathbf{C}_r$ is of size $K \times P_r$, for $r = 1, \ldots, R$. Figure 5.4 shows an example. Bro, Harshman, and Sidiropoulos [33, 6] propose a version of this called the PARALIND model. De Almeida, Favier, and Mota [53] give an overview of some aspects of models of the form in (5.10) and their application to problems in blind beamforming and multiantenna coding. In a series of papers, De Lathauwer [56, 57] and De Lathauwer and Nion [67] (and references therein) explore a general class of decompositions as in (5.10),

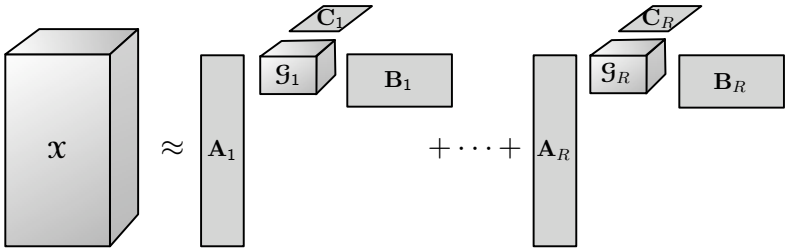


Fig. 5.4: Block decomposition of a third-order tensor.

their uniqueness properties, computational algorithms, and applications in wireless communications; see also [176].

Mahoney et al. [161] extend the matrix CUR decomposition to tensors; others have done related work using sampling methods for tensor approximation [74, 54].

Vasilescu and Terzopoulos [233] explored higher-order versions of independent component analysis (ICA), a variation of PCA that, in some sense, rotates the principal components so that they are statistically independent. Beckmann and Smith [20] extend CP to develop a probabilistic ICA. Bro, Sidiropoulos and Smilde [35] and Vega-Montoto and Wentzell [234] formulate maximum-likelihood versions of CP.

Acar and Yener [6] discuss several of these decompositions as well as some not covered here: shifted CP and Tucker [97] and convoluted CP [171].

6. Software for tensors. The earliest consideration of tensor computation issues dates back to 1973 when Pereyra and Scherer [182] considered basic operations in tensor spaces and how to program them. But general-purpose software for working with tensors has become readily available only in the last decade. In this section, we survey the software available for working with tensors.

MATLAB, Mathematica, and Maple all have support for tensors. MATLAB introduced support for dense multidimensional arrays in version 5.0 (released in 1997). MATLAB supports elementwise manipulation on tensors. More general operations and support for sparse and structured tensors are provided by external toolboxes (N-way Toolbox, CuBatch, PLS_Toolbox, Tensor Toolbox), which are discussed below. Mathematica supports tensors, and there is a Mathematica package for working with tensors that accompanies Ruíz-Tolosa and Castillo [188]. In terms of sparse tensors, Mathematica 6.0 stores its `SparseArray`'s as `Lists`.[4] Maple has the capacity to work with sparse tensors using the `array` command and supports mathematical operations for manipulating tensors that arise in the context of physics and general relativity.

The N-way Toolbox for MATLAB, by Andersson and Bro [11], provides a large collection of algorithms for computing different tensor decompositions. It provides methods for computing CP and Tucker, as well as many of the other models such as multilinear partial least squares (PLS). Additionally, many of the methods can handle constraints (e.g., nonnegativity, orthogonality) and missing data. CuBatch [85] is a graphical user interface in MATLAB for the analysis of data that is built on top of the N-way Toolbox. Its focus is data centric, offering an environment for preprocessing data through diagnostic assessment, such as jack-knifing and bootstrapping. The interface allows custom extensions through an open architecture. Both the N-way

[4] Visit the Mathematica web site (`www.wolfram.com`) and search on "Tensors".

Preprint of article to appear in SIAM Review (June 10, 2008).

Toolbox and CuBatch are freely available.

The commercial PLS_Toolbox for MATLAB [239] also provides a number of multidimensional models, including CP and Tucker, with an emphasis towards data analysis in chemometrics. Like the N-way Toolbox, the PLS_Toolbox can handle constraints and missing data.

The MATLAB Tensor Toolbox, by Bader and Kolda [16, 17, 18], is a general purpose set of classes that extends MATLAB's core capabilities to support operations such as tensor multiplication and matricization. It comes with ALS-based algorithms for CP and Tucker, but the goal is to enable users to easily develop their own algorithms. The Tensor Toolbox is unique in its support for *sparse* tensors, which it stores in coordinate format. Other recommendations for storing sparse tensors have been made; see [156, 155]. The Tensor Toolbox also supports structured tensors so that it can store and manipulate, e.g., a CP representation of a large-scale sparse tensor. The Tensor Toolbox is freely available for research and evaluation purposes.

The Multilinear Engine by Paatero [179] is a FORTRAN code based on the conjugate gradient algorithm that also computes a variety of multilinear models. It supports CP, PARAFAC2, and more.

There are also some packages in C++. The HUJI Tensor Library (HTL) by Zass [243] is a C++ library of classes for tensors, including support for sparse tensors. HTL does not support tensor multiplication, but it does support inner product, addition, elementwise multiplication, etc. FTensor, by Landry [146], is a collection of template-based tensor classes in C++ for general relativity applications; it supports functions such as binary operations and internal and external contractions. The tensors are assumed to be dense, though symmetries are exploited to optimize storage. The Boost Multidimensional Array Library (Boost.MultiArray) [83] provides a C++ class template for multidimensional arrays that is efficient and convenient for expressing dense N-dimensional arrays. These arrays may be accessed using a familiar syntax of native C++ arrays, but it does not have key mathematical concepts for multilinear algebra, such as tensor multiplication.

7. Discussion. This survey has provided an overview of tensor decompositions and their applications. The primary focus was on the CP and Tucker decompositions, but we have also presented some of the other models such as PARAFAC2. There is a flurry of current research on more efficient methods for computing tensor decompositions and better methods for determining typical and maximal tensor ranks.

We have mentioned applications ranging from psychometrics and chemometrics to computer visualization and data mining, but many more applications of tensors are being developed. For instance, in mathematics, Grasedyck [86] uses the tensor structure in finite element computations to generate low-rank Kronecker product approximations. As another example, Lim [154] (see also Qi [183]) discusses higher-order extensions of singular values and eigenvalues. For supersymmetric tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \cdots \times I}$ of order N, the scalar λ is an eigenvalue and the vector $\mathbf{v} \in \mathbb{R}^I$ is its corresponding eigenvector if

$$\mathcal{X} \,\bar{\times}_2\, \mathbf{v} \,\bar{\times}_3\, \mathbf{v} \cdots \,\bar{\times}_N\, \mathbf{v} = \lambda \mathbf{v}.$$

No doubt more applications, both inside and outside of mathematics, will find uses for tensor decompositions in the future.

Thus far, most computational work is done in MATLAB and all is serial. In the future, there will be a need for libraries that take advantage of parallel processors and/or multi-core architectures. Numerous data-centric issues exist as well, such as

preparing the data for processing; see, e.g., [36]. Another problem is missing data, especially systematically missing data, which is an issue in chemometrics; see Tomasi and Bro [222] and references therein. For a general discussion on handling missing data, see Kiers [119].

Acknowledgments. The following workshops have greatly influenced our understanding and appreciation of tensor decompositions: the Workshop on Tensor Decompositions, American Institute of Mathematics, Palo Alto, California, USA, July 19–23, 2004; the Workshop on Tensor Decompositions and Applications (WTDA), CIRM, Luminy, Marseille, France, August 29–September 2, 2005; the workshop on Three-way Methods in Chemistry and Psychology (TRICAP), Chania, Crete, Greece, June 4–9, 2006; and the 24th GAMM Seminar on Tensor Approximations, Max Planck Institute, Leipzig, Germany, January 25–26, 2008. We thank the organizers of these workshops for inviting us.

The following people were kind enough to read earlier versions of this manuscript and offer helpful advice on many fronts: Evrim Acar Ataman, Rasmus Bro, Peter Chew, Lieven De Lathauwer, Lars Eldén, Philip Kegelmeyer, Morten Mørup, Teresa Selee, and Alwin Stegeman. We are also indebted to the anonymous referees, whose feedback was extremely useful in improving the manuscript. Finally, we would like to acknowledge all of our colleauges who have engaged in helpful discussion and provided pointers to related literature, especially: Dario Bini, Pierre Comon, Henk Kiers, Pieter Kroonenberg, J. M. Landsberg, Lek-Heng Lim, Carmeliza Navasca, Berkant Savas, Nikos Sidiropoulos, Jos Ten Berge, and Alex Vasilescu. Finally, we would like to pay special tribute to two colleagues who passed away while this article was in review: Gene Golub and Richard Harshman have directly and profoundly shaped our work and the work of many others. They will be dearly missed.

REFERENCES

[1] E. E. Abdallah, A. B. Hamza, and P. Bhattacharya, *MPEG video watermarking using tensor singular value decomposition*, in Image Analysis and Recognition, vol. 4633 of Lecture Notes in Computer Science, Springer, 2007, pp. 772–783.

[2] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, *Multiway analysis of epilepsy tensors*, Bioinformatics, 23 (2007), pp. i10–i18.

[3] ———, *Seizure recognition on epilepsy feature tensor*, in EMBS 2007: Proceedings of the 29th Annual International Conference on Engineering in Medicine and Biology Society, IEEE, 2007, pp. 4273–4276.

[4] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener, *Modeling and multiway analysis of chatroom tensors*, in ISI 2005: Proceedings of the IEEE International Conference on Intelligence and Security Informatics, vol. 3495 of Lecture Notes in Computer Science, Springer, 2005, pp. 256–268.

[5] E. Acar, S. A. Çamtepe, and B. Yener, *Collective sampling and analysis of high order tensors for chatroom communications*, in ISI 2006: Proceedings of the IEEE International Conference on Intelligence and Security Informatics, vol. 3975 of Lecture Notes in Computer Science, Springer, 2006, pp. 213–224.

[6] E. Acar and B. Yener, *Unsupervised multiway data analysis: a literature survey*. Submitted for publication. Available at `http://www.cs.rpi.edu/research/pdf/07-06.pdf`, 2007.

[7] J. Alexander and A. Hirschowitz, *Polynomical interpolation in several variables*, J. Algebraic Geom., 1995 (1995), pp. 201–222. Cited in [49].

[8] A. H. Andersen and W. S. Rayens, *Structure-seeking multilinear methods for the analysis of fMRI data*, NeuroImage, 22 (2004), pp. 728–739.

[9] C. M. Andersen and R. Bro, *Practical aspects of PARAFAC modeling of fluorescence excitation-emission data*, Journal of Chemometrics, 17 (2003), pp. 200–215.

[10] C. A. Andersson and R. Bro, *Improving the speed of multi-way algorithms: Part I. Tucker3*, Chemometrics and Intelligent Laboratory Systems, 42 (1998), pp. 93–103.

[11] C. A. Andersson and R. Bro, *The N-way toolbox for MATLAB*, Chemometrics and Intelligent Laboratory Systems, 52 (2000), pp. 1–4. See also `http://www.models.kvl.dk/source/nwaytoolbox/`.

[12] C. A. Andersson and R. Henrion, *A general algorithm for obtaining simple structure of core arrays in N-way PCA with application to fluorimetric data*, Computational Statistics & Data Analysis, 31 (1999), pp. 255–278.

[13] C. J. Appellof and E. R. Davidson, *Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents*, Analytical Chemistry, 53 (1981), pp. 2053–2056.

[14] B. W. Bader, M. W. Berry, and M. Browne, *Discussion tracking in enron email using PARAFAC*, in Survey of Text Mining: Clustering, Classification, and Retrieval, Second Edition, M. W. Berry and M. Castellanos, eds., Springer, 2007, pp. 147–162.

[15] B. W. Bader, R. A. Harshman, and T. G. Kolda, *Temporal analysis of semantic graphs using ASALSAN*, in ICDM 2007: Proceedings of the 7th IEEE International Conference on Data Mining, Oct. 2007, pp. 33–42.

[16] B. W. Bader and T. G. Kolda, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Transactions on Mathematical Software, 32 (2006), pp. 635–653.

[17] ———, *Efficient MATLAB computations with sparse and factored tensors*, SIAM Journal on Scientific Computing, 30 (2007), pp. 205–231.

[18] ———, *MATLAB tensor toolbox, version 2.2.* `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`, January 2007.

[19] C. Bauckhage, *Robust tensor classifiers for color object recognition*, in Image Analysis and Recognition, vol. 4633 of Lecture Notes in Computer Science, Springer, 2007, pp. 352–363.

[20] C. Beckmann and S. Smith, *Tensorial extensions of independent component analysis for multisubject FMRI analysis*, NeuroImage, 25 (2005), pp. 294–311.

[21] M. W. Berry, S. T. Dumais, and G. W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Review, 37 (1995), pp. 573–595.

[22] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proceedings of the National Academy of Sciences, 99 (2002), pp. 10246–10251.

[23] ———, *Algorithms for numerical analysis in high dimensions*, SIAM Journal on Scientific Computing, 26 (2005), pp. 2133–2159.

[24] D. Bini, *The role of tensor rank in the complexity analysis of bilinear forms*. Presentation at ICIAM07, Zürich, Switzerland, July 2007.

[25] D. Bini, M. Capovani, G. Lotti, and F. Romani, *$o(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication*, Information Processing Letters, 8 (1979), pp. 234–235.

[26] M. Bläser, *On the complexity of the multiplication of matrices in small formats*, Journal of Complexity, 19 (2003), pp. 43–60. Cited in [24].

[27] R. Bro, *PARAFAC. Tutorial and applications*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 149–171.

[28] ———, *Multi-way analysis in the food industry: Models, algorithms, and applications*, PhD thesis, University of Amsterdam, 1998. Available at `http://www.models.kvl.dk/research/theses/`.

[29] ———, *Review on multiway analysis in chemistry — 2000–2005*, Critical Reviews in Analytical Chemistry, 36 (2006), pp. 279–293.

[30] R. Bro and C. A. Andersson, *Improving the speed of multi-way algorithms: Part II. Compression*, Chemometrics and Intelligent Laboratory Systems, 42 (1998), pp. 105–113.

[31] R. Bro, C. A. Andersson, and H. A. L. Kiers, *PARAFAC2—Part II. Modeling chromatographic data with retention time shifts*, J. Chemometr., 13 (1999), pp. 295–309.

[32] R. Bro and S. De Jong, *A fast non-negativity-constrained least squares algorithm*, Journal of Chemometrics, 11 (1997), pp. 393–401.

[33] R. Bro, R. A. Harshman, and N. D. Sidiropoulos, *Modeling multi-way data with linearly dependent loadings*, Tech. Report 2005-176, KVL, 2005. Cited in [6]. Available at `http://www.telecom.tuc.gr/~nikos/paralind_KVLTechnicalReport2005-176.pdf`.

[34] R. Bro and H. A. L. Kiers, *A new efficient method for determining the number of components in PARAFAC models*, Journal of Chemometrics, 17 (2003), pp. 274–286.

[35] R. Bro, N. Sidiropoulos, and A. Smilde, *Maximum likelihood fitting using ordinary least squares algorithms*, Journal of Chemometrics, 16 (2002), pp. 387–400.

[36] R. Bro and A. K. Smilde, *Centering and scaling in component analysis*, Psychometrika, 17 (2003), pp. 16–33.

[37] N. Bshouty, *Maximal rank of $m \times n \times (mn - k)$ tensors*, SIAM Journal on Computing, 19 (1990), pp. 467–471.

[38] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional*

scaling via an N-way generalization of 'Eckart-Young' decomposition, Psychometrika, 35 (1970), pp. 283–319.

[39] J. D. Carroll, S. Pruzansky, and J. B. Kruskal, *CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters*, Psychometrika, 45 (1980), pp. 3–24.

[40] R. B. Cattell, *Parallel proportional profiles and other principles for determining the choice of factors by rotation*, Psychometrika, 9 (1944), pp. 267–283.

[41] R. B. Cattell, *The three basic factor-analytic research designs — their interrelations and derivatives*, Psychological Bulletin, 49 (1952), pp. 499–452.

[42] E. Ceulemans and H. A. L. Kiers, *Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method*, British Journal of Mathematical and Statistical Psychology, 59 (2006), pp. 133–150.

[43] B. Chen, A. Petropolu, and L. De Lathauwer, *Blind identification of convolutive MIMO systems with 3 sources and 2 sensors*, EURASIP Journal on Applied Signal Processing, (2002), pp. 487–496. (Special Issue on Space-Time Coding and Its Applications, Part II).

[44] P. A. Chew, B. W. Bader, T. G. Kolda, and A. Abdelali, *Cross-language information retrieval using PARAFAC2*, in KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2007, pp. 143–152.

[45] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S.-I. Amari, *Non-negative tensor factorization using alpha and beta divergences*, in ICASSP 07: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 2007.

[46] P. Comon, *Independent component analysis, a new concept?*, Signal Processing, 36 (1994), pp. 287–314.

[47] ———, *Tensor decompositions: State of the art and applications*, in Mathematics in Signal Processing V, J. G. McWhirter and I. K. Proudler, eds., Oxford University Press, 2001, pp. 1–24.

[48] P. Comon, *Canonical tensor decompositions*, I3S Report RR-2004-17, Lab. I3S, 2000 route des Lucioles, B.P.121,F-06903 Sophia-Antipolis cedex, France, June 17 2004.

[49] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SCCM Technical Report 06-02, Stanford University, 2006.

[50] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Processing, 53 (1996), pp. 93–107.

[51] P. Comon, J. M. F. Ten Berge, L. De Lathauwer, and J. Castaing, *Generic and typical ranks of multi-way arrays*. Submitted for publication, 2007.

[52] R. Coppi and S. Bolasco, eds., *Multiway Data Analysis*, North-Holland, Amsterdam, 1989.

[53] A. De Almeida, G. Favier, and J. ao Mota, *The constrained block-PARAFAC decomposition*. Presentation at TRICAP2006, Chania, Greece. Available online at `http://www.telecom.tuc.gr/~nikos/TRICAP2006main/TRICAP2006_Almeida.pdf`, June 2006.

[54] W. F. de la Vega, M. Karpinski, R. Kannan, and S. Vempala, *Tensor decomposition and approximation schemes for constraint satisfaction problems*, in STOC '05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, ACM Press, 2005, pp. 747–754.

[55] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 642–666.

[56] ———, *Decompositions of a higher-order tensor in block terms — Part I: Lemmas for partitioned matrices*. Submitted for publication, 2007.

[57] ———, *Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness*. Submitted for publication, 2007.

[58] L. De Lathauwer and J. Castaing, *Blind identification of underdetermined mixtures by simultaneous matrix diagonalization*. To appear in *IEEE Transactions on Signal Processing.*, 2007.

[59] L. De Lathauwer and J. Castaing, *Tensor-based techniques for the blind separation of DS-CDMA signal*, Signal Processing, 87 (2007), pp. 322–336.

[60] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, *Fourth-order cumulant based blind identification of underdetermined mixtures*, IEEE Transactions on Signal Processing, 55 (2007), pp. 2965–2973.

[61] L. De Lathauwer and A. de Baynast, *Blind deconvolution of DS-CDMA signals by means of decomposition in rank-$(1, l, l)$ terms*. To appear in *IEEE Transactions on Signal Processing.*, 2007.

[62] L. De Lathauwer and B. De Moor, *From matrix to tensor: Multilinear algebra and signal*

processing, in Mathematics in Signal Processing IV, J. McWhirter and I. Proudler, eds., Clarendon Press, Oxford, 1998, pp. 1–15.

[63] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1253–1278.

[64] ———, *On the best rank-1 and rank-($R_1, R_2, \ldots, R_N$) approximation of higher-order tensors*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1324–1342.

[65] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Independent component analysis and (simultaneous) third-order tensor diagonalization*, IEEE Transactions on Signal Processing, 49 (2001), pp. 2262–2271.

[66] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 295–327.

[67] L. De Lathauwer and D. Nion, *Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms*. Submitted for publication, 2007.

[68] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-($R_1, R_2, \ldots, R_N$) reduction in multilinear algebra*, Linear Algebra and its Applications, 391 (2004), pp. 31 – 55.

[69] V. De Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal on Matrix Analysis and Applications, (2006). To appear.

[70] M. De Vos, L. De Lathauwer, B. Vanrumste, S. Van Huffel, and W. Van Paesschen, *Canonical decomposition of ictal scalp EEG and accurate source localisation: Principles and simulation study*, Computational Intelligence and Neuroscience, 2007 (2007), pp. 1–8.

[71] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen, *Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone*, NeuroImage, 37 (2007), pp. 844–854.

[72] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, *Indexing by latent semantic analysis*, Journal of the American Society for Information Science, 41 (1990), pp. 391–407.

[73] A. Doostan, G. Iaccarino, and N. Etemadi, *A least-squares approximation of high-dimensional uncertain systems*, in Annual Research Briefs, Center for Turbulence Research, Stanford University, 2007, pp. 121–132.

[74] P. Drineas and M. W. Mahoney, *A randomized algorithm for a tensor-based generalization of the singular value decomposition*, Linear Algebra and its Applications, 420 (2007), pp. 553–571.

[75] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, *Using latent semantic analysis to improve access to textual information*, in CHI '88: Proceedings of the SIGCHI Conference on Fuman Factors in Computing Systems, ACM Press, 1988, pp. 281–285.

[76] G. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[77] L. Eldén and B. Savas, *A Newton–Grassmann method for computing the best multi-linear rank-(r_1, r_2, r_3) approximation of a tensor*, Tech. Report LITH-MAT-R-2007-6-SE, Department of Mathematics, Linköpings Universitet, 2007.

[78] N. K. M. Faber, R. Bro, and P. K. Hopke, *Recent developments in CANDECOMP/PARAFAC algorithms: A critical review*, Chemometrics and Intelligent Laboratory Systems, 65 (2003), pp. 119–137.

[79] A. S. Field and D. Graupe, *Topographic component (parallel factor) analysis of multichannel evoked potentials: Practical issues in trilinear spatiotemporal decomposition*, Brain Topography, 3 (1991), pp. 407–423.

[80] D. FitzGerald, M. Cranitch, and E. Coyle, *Non-negative tensor factorisation for sound source separation*, in ISSC 2005: Proceedings of the Irish Signals and Systems Conference, 2005.

[81] M. P. Friedlander and K. Hatz, *Computing nonnegative tensor factorizations*, Tech. Report TR-2006-21, Department of Computer Science, University of British Columbia, Oct. 2006.

[82] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, *Appearance based object modeling using texture database: acquisition, compression and rendering*, in EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering, Aire-la-Ville, Switzerland, Switzerland, 2002, Eurographics Association, pp. 257–266.

[83] R. Garcia and A. Lumsdaine, *MultiArray: A C++ library for generic programming with arrays*, Software: Practice and Experience, 35 (2004), pp. 159–188.

[84] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, 1996.

[85] S. Gourvénec, G. Tomasi, C. Durville, E. Di Crescenzo, C. Saby, D. Massart, R. Bro, and G. Oppenheim, *CuBatch, a MATLAB Interface for n-mode Data Analysis*, Chemometrics and Intelligent Laboratory Systems, 77 (2005), pp. 122–130.

[86] L. Grasedyck, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.

[87] V. S. Grigorascu and P. A. Regalia, *Tensor displacement structures and polyspectral matching*, in Fast Reliable Algorithms for Matrices with Structure, T. Kaliath and A. H. Sayed, eds., SIAM, Philadelphia, 1999, pp. 245–276.

[88] W. Hackbusch and B. N. Khoromskij, *Tensor-product approximation to operators and functions in high dimensions*, Journal of Complexity, 23 (2007), pp. 697–714.

[89] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov, *Hierarchical kronecker tensor-product approximations*, Journal of Numerical Mathematics, 13 (2005), pp. 119–156.

[90] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA working papers in phonetics, 16 (1970), pp. 1–84. Available at `http://publish.uwo.ca/~harshman/wpppfac0.pdf`.

[91] ———, *Determination and proof of minimum uniqueness conditions for PARAFAC*, UCLA working papers in phonetics, 22 (1972), pp. 111–117. Available from `http://publish.uwo.ca/~harshman/wpppfac1.pdf`.

[92] ———, *PARAFAC2: Mathematical and technical notes*, UCLA working papers in phonetics, 22 (1972), pp. 30–47.

[93] ———, *Models for analysis of asymmetrical relationships among N objects or stimuli*, in First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology, McMaster University, Hamilton, Ontario, August 1978. Available at `http://publish.uwo.ca/~harshman/asym1978.pdf`.

[94] ———, *An index formulism that generalizes the capabilities of matrix notation and algebra to n-way arrays*, Journal of Chemometrics, 15 (2001), pp. 689–714.

[95] R. A. Harshman, P. E. Green, Y. Wind, and M. E. Lundy, *A model for the analysis of asymmetric data in marketing research*, Market. Sci., 1 (1982), pp. 205–242.

[96] R. A. Harshman and S. Hong, *'Stretch' vs. 'slice' methods for representing three-way structure via matrix notation*, Journal of Chemometrics, 16 (2002), pp. 198–205.

[97] R. A. Harshman, S. Hong, and M. E. Lundy, *Shifted factor analysis-Part I: Models and properties*, J. Chemometr., 17 (2003), pp. 363–378. Cited in [6].

[98] R. A. Harshman and M. E. Lundy, *Data preprocessing and the extended PARAFAC model*, in Research Methods for Multi-mode Data Analysis, H. G. Law, C. W. Snyder, J. Hattie, and R. K. McDonald, eds., Praeger Publishers, New York, 1984, pp. 216–284.

[99] R. A. Harshman and M. E. Lundy, *Three-way DEDICOM: Analyzing multiple matrices of asymmetric relationships*. Paper presented at the Annual Meeting of the North American Psychometric Society, 1992.

[100] R. A. Harshman and M. E. Lundy, *Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP*, Psychometrika, 61 (1996), pp. 133–154.

[101] J. Håstad, *Tensor rank is NP-complete*, Journal of Algorithms, 11 (1990), pp. 644–654.

[102] T. Hazan, S. Polak, and A. Shashua, *Sparse image coding using a 3D non-negative tensor factorization*, in ICCV 2005: Proceedings of the 10th IEEE International Conference on Computer Vision, vol. 1, IEEE Computer Society, 2005, pp. 50–57.

[103] R. Henrion, *Body diagonalization of core matrices in three-way principal components analysis: Theoretical bounds and simulation*, Journal of Chemometrics, 7 (1993), pp. 477–494.

[104] ———, *N-way principal component analysis theory, algorithms and applications*, Chemometrics and Intelligent Laboratory Systems, 25 (1994), pp. 1–23.

[105] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Journal of Mathematics and Physics, 6 (1927), pp. 164–189.

[106] ———, *Multilple invariants and generalized rank of a p-way matrix or tensor*, Journal of Mathematics and Physics, 7 (1927), pp. 39–79.

[107] T. D. Howell, *Global properties of tensor rank*, Linear Algebra and its Applications, 22 (1978), pp. 9–23.

[108] I. Ibraghimov, *Application of the three-way decomposition for matrix compression*, Numerical Linear Algebra with Applications, 9 (2002), pp. 551–565.

[109] J. JáJá, *Optimal evaluation of pairs of bilinear forms*, SIAM Journal on Computing, 8 (1979), pp. 443–462.

[110] J. Jiang, H. Wu, Y. Li, and R. Yu, *Three-way data resolution by alternating slice-wise diagonalization (ASD) method*, Journal of Chemometrics, 14 (2000), pp. 15–36.

[111] T. Jiang and N. Sidiropoulos, *Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints*, IEEE Transactions on Signal Processing, 52 (2004), pp. 2625–2636.

[112] H. Kaiser, *The varimax criterion for analytic rotation in factor analysis*, Psychometrika, 23 (1958), pp. 187–200.

[113] A. Kapteyn, H. Neudecker, and T. Wansbeek, *An approach to n-mode components analysis*, Psychometrika, 51 (1986), pp. 269–275.

[114] B. N. Khoromskij, *Structured rank-$(r_1, \ldots, r_d)$ decomposition of function-related operators in r^d*, Computational Methods in Applied Mathematics, 6 (2006), pp. 194–220.

[115] B. N. Khoromskij and V. Khoromskaia, *Low rank Tucker-type tensor approximation to classical potentials*, Central European Journal of Mathematics, 5 (2007), pp. 523–550.

[116] H. A. L. Kiers, *An alternating least squares algorithm for fitting the two- and three-way DEDICOM model and the IDIOSCAL model*, Psychometrika, 54 (1989), pp. 515–521.

[117] ———, *TUCKALS core rotations and constrained TUCKALS modelling*, Statistica Applicata, 4 (1992), pp. 659–667.

[118] ———, *An alternating least squares algorithm for PARAFAC2 and three-way DEDICOM*, Comput. Stat. Data. An., 16 (1993), pp. 103–118.

[119] ———, *Weighted least squares fitting using ordinary least squares algorithms*, Psychometrika, 62 (1997), pp. 215–266.

[120] ———, *Joint orthomax rotation of the core and component matrices resulting from three-mode principal components analysis*, Journal of Classification, 15 (1998), pp. 245 – 263.

[121] ———, *A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity*, Journal of Chemometrics, 12 (1998), pp. 255–171.

[122] ———, *Towards a standardized notation and terminology in multiway analysis*, Journal of Chemometrics, 14 (2000), pp. 105–122.

[123] H. A. L. Kiers and A. Der Kinderen, *A fast method for choosing the numbers of components in Tucker3 analysis*, British Journal of Mathematical and Statistical Psychology, 56 (2003), pp. 119–125.

[124] H. A. L. Kiers and R. A. Harshman, *Relating two proposed methods for speedup of algorithms for fitting two- and three-way principal component and related multilinear models*, Chemometrics and Intelligent Laboratory Systems, 36 (1997), pp. 31–40.

[125] H. A. L. Kiers and Y. Takane, *Constrained DEDICOM*, Psychometrika, 58 (1993), pp. 339–355.

[126] H. A. L. Kiers, J. M. F. Ten Berge, and R. Bro, *PARAFAC2—Part I. A direct fitting algorithm for the PARAFAC2 model*, Journal of Chemometrics, 13 (1999), pp. 275–294.

[127] H. A. L. Kiers, J. M. F. ten Berge, and R. Rocci, *Uniqueness of three-mode factor models with sparse cores: The $3 \times 3 \times 3$ case*, Psychometrika, 62 (1997), pp. 349–374.

[128] H. A. L. Kiers, J. M. F. ten Berge, Y. Takane, and J. de Leeuw, *A generalization of Takane's algorithm for DEDICOM*, Psychometrika, 55 (1990), pp. 151–158.

[129] H. A. L. Kiers and I. Van Mechelen, *Three-way component analysis: Principles and illustrative application*, Psychological Methods, 6 (2001), pp. 84–110.

[130] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, vol. 2, Addison-Wesley, 3rd ed., 1997.

[131] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM Journal on Matrix Analysis and Applications, 23 (2002), pp. 863–884.

[132] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 243–255.

[133] ———, *A counterexample to the possibility of an extension of the Eckart-Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM Journal on Matrix Analysis and Applications, 24 (2003), pp. 762–767.

[134] ———, *Multilinear operators for higher-order decompositions*, Tech. Report SAND2006-2081, Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, Apr. 2006.

[135] T. G. Kolda and B. W. Bader, *The TOPHITS model for higher-order web link analysis*, in Workshop on Link Analysis, Counterterrorism and Security, 2006.

[136] T. G. Kolda, B. W. Bader, and J. P. Kenny, *Higher-order web link analysis using multilinear algebra*, in ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining, IEEE Computer Society, 2005, pp. 242–249.

[137] W. P. Krijnen, *Convergence of the sequence of parameters generated by alternating least squares algorithms*, Computational Statistics & Data Analysis, 51 (2006), pp. 481–489. Cited by [205].

[138] P. M. Kroonenberg, *Three-Mode Principal Component Analysis: Theory and Applications*, DWO Press, Leiden, The Netherlands, 1983. Available at `http://three-mode.leidenuniv.nl/bibliogr/kroonenbergpm_thesis/index.html`.

[139] ———, *Applied Multiway Data Analysis*, Wiley, 2008.

[140] P. M. Kroonenberg and J. De Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.

[141] J. B. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra and its Applications, 18 (1977), pp. 95–138.

[142] ———, *Statement of some current results about three-way arrays*. Unpublished manuscript, AT&T Bell Laboratories, Murray Hill, NJ. Available at `http://three-mode.leidenuniv.nl/pdf/k/kruskal1983.pdf`, 1983.

[143] J. B. Kruskal, *Rank, decomposition, and uniqueness for 3-way and N-way arrays*, in Coppi and Bolasco [52], pp. 7–18.

[144] J. B. Kruskal, R. A. Harshman, and M. E. Lundy, *How 3-MFA can cause degenerate PARAFAC solutions, among other relationships*, in Coppi and Bolasco [52], pp. 115–121.

[145] J. D. Laderman, *A noncommutative algorithm for multiplying* 3×3 *matrices using 23 multiplications*, Bulletin of the American Mathematical Society, 82 (1976), pp. 126–128. Cited in [24].

[146] W. Landry, *Implementing a high performance tensor library*, Scientific Programming, 11 (2003), pp. 273–290.

[147] J. M. Landsberg, *The border rank of the multiplication of* 2×2 *matrices is seven*, Journal of the American Mathematical Society, 19 (2005), pp. 447–459.

[148] ———, *Geometry and the complexity of matrix multiplication*. Available at `http://www.math.tamu.edu/~jml/msurvey0407.pdf`. To appear in *Bull. AMS.*, 2007.

[149] A. N. Langville and W. J. Stewart, *A Kronecker product approximate precondition for SANs*, Numerical Linear Algebra with Applications, 11 (2004), pp. 723–752.

[150] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.

[151] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[152] S. Leurgans and R. T. Ross, *Multilinear models: Applications in spectroscopy*, Statistical Science, 7 (1992), pp. 289–310.

[153] J. Levin, *Three-mode factor analysis*, PhD thesis, University of Illinois, Urbana, 1963. Cited in [226, 140].

[154] L.-H. Lim, *Singular values and eigenvalues of tensors: A variational approach*, in CAMSAP2005: 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005, pp. 129–132.

[155] C.-Y. Lin, Y.-C. Chung, and J.-S. Liu, *Efficient data compression methods for multidimensional sparse array operations based on the EKMR scheme*, IEEE Transactions on Computers, 52 (2003), pp. 1640–1646.

[156] C.-Y. Lin, J.-S. Liu, and Y.-C. Chung, *Efficient representation scheme for multidimensional array operations*, IEEE Transactions on Computers, 51 (2002), pp. 327–345.

[157] N. Liu, B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien, *Text representation: From vector to tensor*, in ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining, IEEE Computer Society, 2005, pp. 725–728.

[158] X. Liu and N. Sidiropoulos, *Cramér-Rao lower bounds for low-rank decomposition of multidimensional arrays*, IEEE Transactions on Signal Processing, 49 (2001), pp. 2074–2086.

[159] M. E. Lundy, R. A. Harshman, and J. B. Kruskal, *A two-stage procedure incorporating good features of both trilinear and quadrilinear methods*, in Coppi and Bolasco [52], pp. 123–130.

[160] M. E. Lundy, R. A. Harshman, P. Paatero, and L. Swartzman, *Application of the 3-way DEDICOM model to skew-symmetric data for paired preference ratings of treatments for chronic back pain*. Presentation at TRICAP2003, Lexington, Kentucky, June 2003. Available at `http://publish.uwo.ca/~harshman/tricap03.pdf`.

[161] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, in KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2006, pp. 327–336.

[162] C. D. M. Martin and C. F. Van Loan, *A Jacobi-type method for computing orthogonal tensor decompositions*. To appear in SIAM Journal on Matrix Analysis and Applications. Available from `http://www.cam.cornell.edu/~carlam/academic/JACOBI3way_paper.pdf`., 2006.

[163] E. Martínez-Montes, P. A. Valdés-Sosa, F. Miwakeichi, R. I. Goldman, and M. S.

Cohen, *Concurrent EEG/fMRI analysis by multiway partial least squares*, NeuroImage, 22 (2004), pp. 1023–1034.

[164] B. C. Mitchell and D. S. Burdick, *Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies*, Journal of Chemometrics, 8 (1994), pp. 155–168.

[165] F. Miwakeichi, E. Martínez-Montes, P. A. Valds-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, *Decomposing EEG data into space-time-frequency components using parallel factor analysis*, NeuroImage, 22 (2004), pp. 1035–1045.

[166] J. Möcks, *Topographic components model for event-related potentials and some biophysical considerations*, IEEE Transactions on Biomedical Engineering, 35 (1988), pp. 482–484.

[167] M. Mørup, L. Hansen, J. Parnas, and S. M. Arnfred, *Decomposing the time-frequency representation of EEG using nonnegative matrix and multi-way factorization*. Available at `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4144/pdf/imm4144.pdf`, 2006.

[168] M. Mørup, L. K. Hansen, and S. M. Arnfred, *Algorithms for sparse non-negative TUCKER (also named HONMF)*, Tech. Report IMM4658, Technical University of Denmark, 2006.

[169] M. Mørup, L. K. Hansen, and S. M. Arnfred, *ERPWAVELAB a toolbox for multi-channel analysis of time-frequency transformed event related potentials*, Journal of Neuroscience Methods, 161 (2007), pp. 361–368.

[170] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, *Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG*, NeuroImage, 29 (2006), pp. 938–947.

[171] M. Mørup, M. N. Schmidt, and L. K. Hansen, *Shift invariant sparse coding of image and music data*, tech. report, Technical University of Denmark, 2007. Cited in [6].

[172] T. Murakami, J. M. F. Ten Berge, and H. A. L. Kiers, *A case of extreme simplicity of the core matrix in three-mode principal components analysis*, Psychometrika, 63 (1998), pp. 255–261.

[173] D. Muti and S. Bourennane, *Multidimensional filtering based on a tensor approach*, Signal Processing, 85 (2005), pp. 2338–2353.

[174] J. G. Nagy and M. E. Kilmer, *Kronecker product approximation in three-dimensional imaging applications*, IEEE Transactions on Image Processing, 15 (2006), pp. 604–613.

[175] M. N. L. Narasimhan, *Principles of Continuum Mechanics*, John Wiley & Sons, New York, 1993.

[176] C. Navasca, L. De Lathauwer, and S. Kinderman, *Swamp reducing technique for tensor decompositions*. Submitted for publication, Mar. 2008.

[177] D. Nion and L. De Lathauwer, *An enhanced line search scheme for complex-valued tensor decompositions. application in DS-CDMA*, Signal Processing, 88 (2008), pp. 749–755.

[178] P. Paatero, *A weighted non-negative least squares algorithm for three-way "PARAFAC" factor analysis*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 223–242.

[179] ———, *The multilinear engine: A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model*, Journal of Computational and Graphical Statistics, 8 (1999), pp. 854–888.

[180] ———, *Construction and analysis of degenerate PARAFAC models*, Journal of Chemometrics, 14 (2000), pp. 285–299.

[181] P. Paatero and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[182] V. Pereyra and G. Scherer, *Efficient computer manipulation of tensor products with applications to multidimensional approximation*, Mathematics of Computation, 27 (1973), pp. 595–605.

[183] L. Qi, *Eigenvalues of a real supersymmetric tensor*, J. Symb. Comput., 40 (2005), pp. 1302–1324. Cited by [154].

[184] L. Qi, W. Sun, and Y. Wang, *Numerical multilinear algebra and its applications*, Frontiers of Mathematics in China, 2 (2007), pp. 501–526.

[185] M. Rajih and P. Comon, *Enhanced line search: A novel method to accelerate PARAFAC*, in EUSIPCO-05: Proceedings of the 13th European Signal Processing Conference, 2005.

[186] W. S. Rayens and B. C. Mitchell, *Two-factor degeneracies and a stabilization of PARAFAC*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), p. 173.

[187] R. Rocci, *A general algorithm to fit constrained DEDICOM models*, Statistical Methods and Applications, 13 (2004), pp. 139–150.

[188] J. R. Ruíz-Tolosa and E. Castillo, *From Vectors to Tensors*, Universitext, Springer, Berlin,

2005.

[189] E. Sanchez and B. R. Kowalski, *Tensorial resolution: A direct trilinear decomposition*, Journal of Chemometrics, 4 (1990), pp. 29–45.

[190] B. Savas, *Analyses and tests of handwritten digit recognition algorithms*, master's thesis, Linköping University, Sweden, 2003.

[191] B. Savas and L. Eldén, *Handwritten digit classification using higher order singular value decomposition*, Pattern Recognition, 40 (2007), pp. 993–1003.

[192] A. Schönhage, *Partial and total matrix multiplication*, SIAM Journal on Computing, 10 (1981), pp. 434–455. Cited in [24, 148].

[193] A. Shashua and T. Hazan, *Non-negative tensor factorization with applications to statistics and computer vision*, in ICML 2005: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 792–799.

[194] A. Shashua and A. Levin, *Linear image coding for regression and classification using the tensor-rank principle*, in CVPR 2001: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. 42–49.

[195] A. Shashua, R. Zass, and T. Hazan, *Multi-way clustering using super-symmetric non-negative tensor factorization*, in ECCV 2006: Proceedings of the 9th European Conference on Computer Vision, Part IV, 2006, pp. 595–608.

[196] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Transactions on Signal Processing, 48 (2000), pp. 2377–2388.

[197] N. Sidiropoulos and R. Budampati, *Khatri-Rao space-time codes*, IEEE Transactions on Signal Processing, 50 (2002), pp. 2396–2407.

[198] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Transactions on Signal Processing, 48 (2000), pp. 810–823.

[199] N. D. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of N-way arrays*, Journal of Chemometrics, 14 (2000), pp. 229–239.

[200] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*, Wiley, West Sussex, England, 2004.

[201] A. K. Smilde, Y. Wang, and B. R. Kowalski, *Theory of medium-rank second-order calibration with restricted-Tucker models*, Journal of Chemometrics, 8 (1994), pp. 21–36.

[202] A. Stegeman, *Degeneracy in CANDECOMP/PARAFAC explained for $p \times p \times 2$ arrays of rank $p+1$ or higher*, Psychometrika, 71 (2006), pp. 483–501.

[203] ———, *Comparing independent component analysis and the PARAFAC model for artificial multi-subject fMRI data*, tech. report, Heymans Institute for Psychological Research, University of Groningen, The Netherlands, Feb. 2007.

[204] ———, *Degeneracy in Candecomp/Parafac and Indscal explained for several three-sliced arrays with a two-valued typical rank*, Psychometrika, (2007). To appear.

[205] A. Stegeman and L. De Lathauwer, *A method to avoid "Degeneracy" in the CANDECOMP/PARAFAC model for generic $I \times J \times 2$ arrays*. Submitted for publication, 2007.

[206] A. Stegeman and N. D. Sidiropoulos, *On Kruskal's uniqueness condition for the CANDECOMP/PARAFAC decomposition*, Linear Algebra and its Applications, 420 (2007), pp. 540–552.

[207] A. Stegeman, J. M. F. Ten Berge, and L. De Lathauwer, *Sufficient conditions for uniqueness in CANDECOMP/PARAFAC and INDSCAL with random component matrices*, Psychometrika, 71 (2006), pp. 219–229.

[208] V. Strassen, *Gaussian elimination is not optimal*, Numerische Mathematik, 13 (1969), pp. 354–356.

[209] J. Sun, S. Papadimitriou, and P. S. Yu, *Window-based tensor analysis on high-dimensional and multi-aspect streams*, in ICDM 2006: Proceedings of the 6th IEEE Conference on Data Mining, IEEE Computer Society, 2006, pp. 1076–1080.

[210] J. Sun, D. Tao, and C. Faloutsos, *Beyond streams and graphs: Dynamic tensor analysis*, in KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2006, pp. 374–383.

[211] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, *CubeSVD: A novel approach to personalized Web search*, in WWW 2005: Proceedings of the 14th International Conference on World Wide Web, ACM Press, 2005, pp. 382–390.

[212] J. M. F. Ten Berge, *Kruskal's polynomial for $2\times2\times2$ arrays and a generalization to $2\times n\times n$ arrays*, Psychometrika, 56 (1991), pp. 631–636.

[213] ———, *The typical rank of tall three-way arrays*, Psychometrika, 65 (2000), pp. 525–532.

[214] ———, *Partial uniqueness in CANDECOMP/PARAFAC*, Journal of Chemometrics, 18 (2004), pp. 12–16.

[215] J. M. F. ten Berge and H. A. L. Kiers, *Some uniqueness results for PARAFAC2*, J.

Chemometr., 61 (1996), pp. 123–132.

[216] J. M. F. Ten Berge and H. A. L. Kiers, *Simplicity of core arrays in three-way principal component analysis and the typical rank of $p \times q \times 2$ arrays*, Linear Algebra and its Applications, 294 (1999), pp. 169–179.

[217] J. M. F. Ten Berge, H. A. L. Kiers, and J. de Leeuw, *Explicit CANDECOMP/PARAFAC solutions for a contrived $2 \times 2 \times 2$ array of rank three*, Psychometrika, 53 (1988), pp. 579–583.

[218] J. M. F. Ten Berge and N. D. Sidiriopolous, *On uniqueness in CANDECOMP/PARAFAC*, Psychometrika, 67 (2002), pp. 399–409.

[219] J. M. F. Ten Berge, N. D. Sidiropoulos, and R. Rocci, *Typical rank and indscal dimensionality for symmetric three-way arrays of order $I \times 2 \times 2$ or $I \times 3 \times 3$*, Linear Algebra and its Applications, 388 (2004), pp. 363–377.

[220] G. Tomasi, *Use of the properties of the Khatri-Rao product for the computation of Jacobian, Hessian, and gradient of the PARAFAC model under MATLAB*, 2005. Private communication.

[221] ———, *Practical and computational aspects in chemometric data analysis*, PhD thesis, Department of Food Science, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark, 2006. Available at `http://www.models.life.ku.dk/research/theses/`.

[222] G. Tomasi and R. Bro, *PARAFAC and missing values*, Chemometrics and Intelligent Laboratory Systems, 75 (2005), pp. 163–180.

[223] ———, *A comparison of algorithms for fitting the PARAFAC model*, Computational Statistics & Data Analysis, 50 (2006), pp. 1700–1734.

[224] L. R. Tucker, *Implications of factor analysis of three-way matrices for measurement of change*, in Problems in Measuring Change, C. W. Harris, ed., University of Wisconsin Press, 1963, pp. 122–137. Cited in [226, 140].

[225] ———, *The extension of factor analysis to three-dimensional matrices*, in Contributions to Mathematical Psychology, H. Gulliksen and N. Frederiksen, eds., Holt, Rinehardt, & Winston, New York, 1964. Cited in [226, 140].

[226] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[227] C. F. Van Loan, *The ubiquitous Kronecker product*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 85–100.

[228] M. A. O. Vasilescu, *Human motion signatures: Analysis, synthesis, recognition*, in ICPR 2002: Proceedings of the 16th International Conference on Pattern Recognition, IEEE Computer Society, 2002, p. 30456.

[229] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear analysis of image ensembles: TensorFaces*, in ECCV 2002: Proceedings of the 7th European Conference on Computer Vision, vol. 2350 of Lecture Notes in Computer Science, Springer, 2002, pp. 447–460.

[230] ———, *Multilinear image analysis for facial recognition*, in ICPR 2002: Proceedings of the 16th International Conference on Pattern Recognition, 2002, pp. 511– 514.

[231] ———, *Multilinear subspace analysis of image ensembles*, in CVPR 2003: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2003, pp. 93–99.

[232] M. A. O. Vasilescu and D. Terzopoulos, *Tensortextures: multilinear image-based rendering*, ACM Transactions on Graphics, 23 (2004), pp. 336–342.

[233] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear independent components analysis*, in CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2005.

[234] L. Vega-Montoto and P. D. Wentzell, *Maximum likelihood parallel factor analysis (MLPARAFAC)*, Journal of Chemometrics, 17 (2003), pp. 237–253.

[235] D. Vlasic, M. Brand, H. Pfister, and J. Popović, *Face transfer with multilinear models*, ACM Transactions on Graphics, 24 (2005), pp. 426–433.

[236] H. Wang and N. Ahuja, *Facial expression decomposition*, in ICCV 2003: Proceedings of the 9th IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 958–965.

[237] ———, *Compact representation of multidimensional data using tensor rank-one decomposition*, in ICPR 2004: Proceedings of the 17th Interational Conference on Pattern Recognition, vol. 1, 2004, pp. 44–47.

[238] M. Welling and M. Weber, *Positive tensor factorization*, Pattern Recognition Letters, 22 (2001), pp. 1255–1261.

[239] B. M. Wise and N. B. Gallagher, *PLS_Toolbox 4.0*. `http://www.eigenvector.com`, 2007.

[240] B. M. Wise, N. B. Gallagher, and E. B. Martin, *Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch*, Journal of Chemometrics, 15 (2001), pp. 285–

298.

[241] H.-L. Wu, M. Shibukawa, and K. Oguma, *An alternating trilinear decomposition algorithm with application to calibration of HPLC–DAD for simultaneous determination of overlapped chlorinated aromatic hydrocarbons*, Journal of Chemometrics, 12 (1998), pp. 1–26.

[242] E. Zander and H. G. Matthies, *Tensor product methods for stochastic problems.* Preprint obtained via private communication, 2007.

[243] R. Zass, *HUJI tensor library.* `http://www.cs.huji.ac.il/~zass/htl/`, May 2006.

[244] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 534–550.