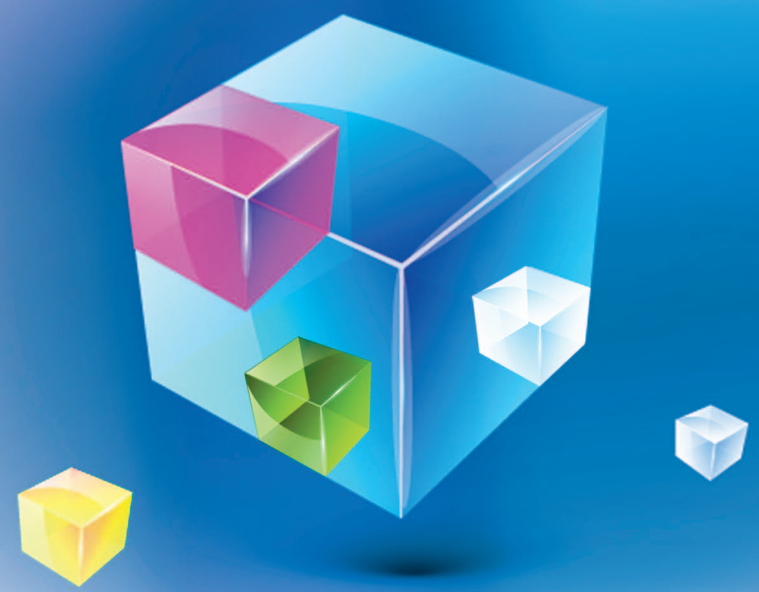
Andrzej Cichocki, Danilo P. Mandic,
Anh Huy Phan, Cesar F. Caiafa,
Guoxu Zhou, Qibin Zhao, and
Lieven De Lathauwer



TENSOR DECOMPOSITIONS for Signal Processing Applications

[From two-way to multiway component analysis]

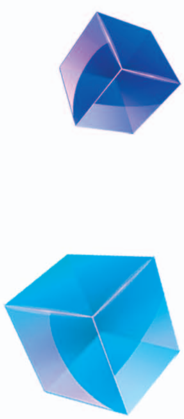


IMAGE LICENSED BY GRAPHIC STOCK

Digital Object Identifier 10.1109/MSP.2013.2297439
Date of publication: 12 February 2015

The widespread use of multisensor technology and the emergence of big data sets have highlighted the limitations of standard flat-view matrix models and the necessity to move toward more versatile data analysis tools. We show that higher-order tensors (i.e., multiway arrays) enable such a fundamental paradigm shift toward models that are essentially polynomial, the uniqueness of which, unlike the matrix methods, is guaranteed under very mild and natural conditions. Benefiting from the power of multilinear algebra as their mathematical backbone, data analysis techniques using tensor decompositions are shown to have great flexibility in the choice of constraints which match data properties and extract more general latent components in the data than matrix-based methods.

A comprehensive introduction to tensor decompositions is provided from a signal processing perspective, starting from the algebraic foundations, via basic canonical polyadic and Tucker models, to advanced cause-effect and multiview data analysis schemes. We show that tensor decompositions enable natural generalizations of some commonly used signal processing paradigms, such as canonical correlation and subspace techniques, signal separation, linear regression, feature extraction, and classification. We also cover computational aspects and point out how ideas from compressed sensing (CS) and scientific computing may be used for addressing the otherwise unmanageable storage and manipulation issues associated with big data sets. The concepts are supported by illustrative real-world case studies that highlight the benefits of the tensor framework as efficient and promising tools, inter alia, for modern signal processing, data analysis, and machine-learning applications; moreover, these benefits also extend to vector/matrix data through tensorization.

HISTORICAL NOTES

The roots of multiway analysis can be traced back to studies of homogeneous polynomials in the 19th century, with contributors including Gauss, Kronecker, Cayley, Weyl, and Hilbert. In the modern-day interpretation, these are fully symmetric tensors. Decompositions of nonsymmetric tensors have been studied since the early 20th century [1], whereas the benefits of using more than two matrices in factor analysis (FA) [2] have been apparent in several communities since the 1960s. The Tucker decomposition (TKD) for tensors was introduced in psychometrics [3], [4], while the canonical polyadic decomposition (CPD) was independently rediscovered and put into an application context under the names of canonical decomposition (CANDECOMP) in psychometrics [5] and parallel factor model (PARAFAC) in linguistics [6]. Tensors were subsequently adopted in diverse branches of data analysis such as chemometrics, the food industry, and social sciences [7], [8]. When it comes to signal processing, the early 1990s saw a considerable interest in higher-order statistics (HOS) [9], and it was soon realized that, for multivariate cases, HOS are effectively higher-order tensors; indeed, algebraic approaches to independent component analysis (ICA) using HOS [10]–[12] were inherently tensor based. Around 2000, it was realized that the TKD represents a multilinear singular value decomposition (MLSVD) [15]. Generalizing the matrix singular value decomposition (SVD), the workhorse of numerical linear algebra, the MLSVD spurred the interest in tensors in applied mathematics and scientific computing in very high dimensions [16]–[18]. In parallel, CPD was successfully adopted as a tool for sensor array processing and deterministic signal separation in wireless communication [19], [20]. Subsequently, tensors have been used in audio, image and video processing, machine learning, and biomedical applications, to name but a few areas. The significant interest in tensors and their quickly emerging applications is reflected in books [7], [8], [12], [21]–[23] and tutorial papers [24]–[31] covering various aspects of multiway analysis.

FROM A MATRIX TO A TENSOR

Approaches to two-way (matrix) component analysis are well established and include principal component analysis (PCA), ICA, nonnegative matrix factorization (NMF), and sparse component analysis (SCA) [12], [21], [32]. These techniques have become standard tools for, e.g., blind source separation (BSS), feature extraction, or classification. On the other hand, large classes of data arising from modern heterogeneous sensor modalities have a multiway character and are, therefore, naturally represented by multiway arrays or tensors (see the section "Tensorization—Blessing of Dimensionality").

Early multiway data analysis approaches reformatted the data tensor as a matrix and resorted to methods developed for classical two-way analysis. However, such a flattened view of the world and the rigid assumptions inherent in two-way analysis are not always a good match for multiway data. It is only through higher-order tensor decomposition that we have the opportunity to develop sophisticated models capturing multiple interactions and couplings instead of standard pairwise interactions. In other words, we can only discover hidden components within multiway data if the analysis tools account for the intrinsic multidimensional patterns present, motivating the development of multilinear techniques.

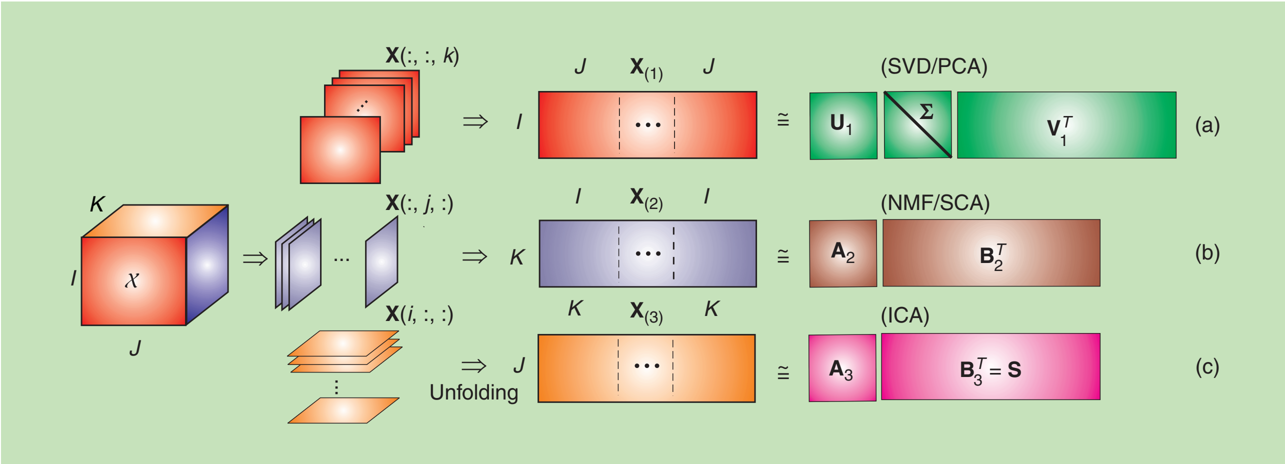
In this article, we emphasize that tensor decompositions are not just matrix factorizations with additional subscripts, multilinear algebra is much more structurally rich than linear algebra. For example, even basic notions such as rank have a more subtle meaning, the uniqueness conditions of higher-order tensor decompositions are more relaxed and accommodating than those for matrices [33], [34], while matrices and tensors also have completely different geometric properties [22]. This boils down to matrices representing linear transformations and quadratic forms, while tensors are connected with multilinear mappings and multivariate polynomials [31].

NOTATIONS AND CONVENTIONS

A tensor can be thought of as a multi-index numerical array, whereby the order of a tensor is the number of its modes or

[TABLE 1] BASIC NOTATION.

$\mathcal{A}, \mathbf{A}, \boldsymbol{a}, a$	TENSOR, MATRIX, VECTOR, SCALAR
$\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R]$	MATRIX $\mathbf{A}$ WITH COLUMN VECTORS $\boldsymbol{a}_r$
$\boldsymbol{a}(:, i_2, i_3, \ldots, i_N)$	FIBER OF TENSOR $\mathcal{A}$ OBTAINED BY FIXING ALL BUT ONE INDEX
$\mathbf{A}(:, :, i_3, \ldots, i_N)$	MATRIX SLICE OF TENSOR $\mathcal{A}$ OBTAINED BY FIXING ALL BUT TWO INDICES
$\mathcal{A}(:, :, :, i_4, \ldots, i_N)$	TENSOR SLICE OF $\mathcal{A}$ OBTAINED BY FIXING SOME INDICES
$\mathcal{A}(\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N)$	SUBTENSOR OF $\mathcal{A}$ OBTAINED BY RESTRICTING INDICES TO BELONG TO SUBSETS $\mathcal{I}_n \subseteq \{1, 2, \ldots, I_n\}$
$\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$	MODE-n MATRICIZATION OF TENSOR $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ WHOSE ENTRY AT ROW i_n AND COLUMN $(i_1 - 1) I_2 \cdots I_{n-1} I_{n+1} \cdots I_N + \cdots + (i_{N-1} - 1) I_N + i_N$ IS EQUAL TO $a_{i_1 i_2 \ldots i_N}$
$\mathrm{vec}(\mathcal{A}) \in \mathbb{R}^{I_N I_{N-1} \cdots I_1}$	VECTORIZATION OF TENSOR $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ WITH THE ENTRY AT POSITION $i_1 + \sum_{k=2}^{N} [(i_k - 1) I_1 I_2 \cdots I_{k-1}]$ EQUAL TO $a_{i_1 i_2 \ldots i_N}$
$\mathbf{D} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_R)$	DIAGONAL MATRIX WITH $d_{rr} = \lambda_r$
$\mathcal{D} = \mathrm{diag}_N(\lambda_1, \lambda_2, \ldots, \lambda_R)$	DIAGONAL TENSOR OF ORDER N WITH $d_{rr \cdots r} = \lambda_r$
$\mathbf{A}^T, \mathbf{A}^{-1}, \mathbf{A}^{\dagger}$	TRANSPOSE, INVERSE, AND MOORE–PENROSE PSEUDOINVERSE



[FIG1] MWCA for a third-order tensor, assuming that the components are (a) principal and orthogonal in the first mode, (b) nonnegative and sparse in the second mode, and (c) statistically independent in the third mode.

dimensions; these may include space, time, frequency, trials, classes, and dictionaries. A real-valued tensor of order N is denoted by $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and its entries by $a_{i_1, i_2, \ldots, i_N}$. Then, an $N \times 1$ vector $\boldsymbol{a}$ is considered a tensor of order one, and an $N \times M$ matrix $\mathbf{A}$ a tensor of order two. Subtensors are parts of the original data tensor, created when only a fixed subset of indices is used. Vector-valued subtensors are called *fibers*, defined by fixing every index but one, and matrix-valued subtensors are called *slices*, obtained by fixing all but two indices (see Table 1). The manipulation of tensors often requires their reformatting (*reshaping*); a particular case of reshaping tensors to matrices is termed *matrix unfolding* or *matricization* (see Figure 1). Note that a mode-n multiplication of a tensor $\mathcal{A}$ with a matrix $\mathbf{B}$ amounts to the multiplication of all mode-n vector fibers with $\mathbf{B}$, and that, in linear algebra, the tensor (or outer) product appears in the expression for a rank-1 matrix: $\boldsymbol{a}\boldsymbol{b}^T = \boldsymbol{a} \circ \boldsymbol{b}$. Basic tensor notations are summarized in Table 1, various product rules used in this article are given in Table 2, while Figure 2 shows two particular ways to construct a tensor.

INTERPRETABLE COMPONENTS IN TWO-WAY DATA ANALYSIS

The aim of BSS, FA, and latent variable analysis is to decompose a data matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ into the factor matrices $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R] \in \mathbb{R}^{I \times R}$ and $\mathbf{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_R] \in \mathbb{R}^{J \times R}$ as

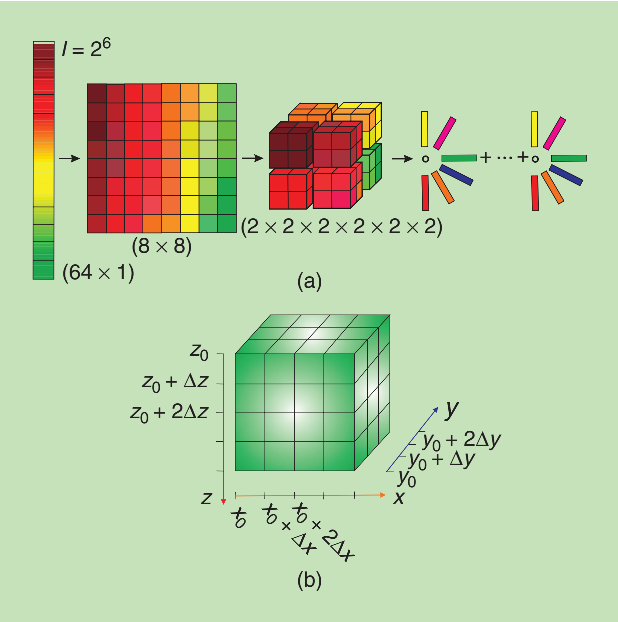
$$\mathbf{X} = \mathbf{A}\mathbf{D}\mathbf{B}^T + \mathbf{E} = \sum_{r=1}^{R} \lambda_r \, \boldsymbol{a}_r \boldsymbol{b}_r^T + \mathbf{E} = \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r + \mathbf{E}, \quad (1)$$

where $\mathbf{D} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_R)$ is a scaling (normalizing) matrix, the columns of $\mathbf{B}$ represent the unknown source signals (factors or latent variables depending on the tasks in hand), the columns of $\mathbf{A}$ represent the associated mixing vectors (or factor loadings), while $\mathbf{E}$ is noise due to an unmodeled data part or model error. In other words, model (1) assumes that the data matrix $\mathbf{X}$ comprises hidden components $\boldsymbol{b}_r$ $(r = 1, 2, \ldots, R)$ that are mixed together in an unknown manner through coefficients $\mathbf{A}$, or, equivalently, that data contain factors that have an associated loading for every data channel. Figure 3(a) depicts the model (1) as a dyadic decomposition, whereby the terms $\boldsymbol{a}_r \circ \boldsymbol{b}_r = \boldsymbol{a}_r \boldsymbol{b}_r^T$ are rank-1 matrices.

The well-known indeterminacies intrinsic to this model are: 1) arbitrary scaling of components and 2) permutation of the rank-1 terms. Another indeterminacy is related to the physical meaning of the factors: if the model in (1) is unconstrained, it admits infinitely many combinations of $\mathbf{A}$ and $\mathbf{B}$. Standard matrix factorizations in linear algebra, such as QR-factorization, eigenvalue decomposition (EVD), and SVD, are only special

[TABLE 2] DEFINITION OF PRODUCTS.

$\mathcal{C} = \mathcal{A} \times_n \mathbf{B}$	MODE-n PRODUCT OF $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ AND $\mathbf{B}\in\mathbb{R}^{J_n\times I_n}$ YIELDS $\mathcal{C}\in\mathbb{R}^{I_1\times\cdots\times I_{n-1}\times J_n\times I_{n+1}\times\cdots\times I_N}$ WITH ENTRIES $c_{i_1\cdots i_{n-1}\, j_n\, i_{n+1}\cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1\cdots i_{n-1} i_n i_{n+1}\cdots i_N}\, b_{j_n i_n}$ AND MATRIX REPRESENTATION $\mathbf{C}_{(n)} = \mathbf{B}\mathbf{A}_{(n)}$
$\mathcal{C} = [\![\mathcal{A}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!]$	FULL MULTILINEAR PRODUCT, $\mathcal{C} = \mathcal{A}\times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)}$
$\mathcal{C}=\mathcal{A}\circ\mathcal{B}$	TENSOR OR OUTER PRODUCT OF $\mathcal{A}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ AND $\mathcal{B}\in\mathbb{R}^{J_1\times J_2\times\cdots\times J_M}$ YIELDS $\mathcal{C}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N\times J_1\times J_2\times\cdots\times J_M}$ WITH ENTRIES $c_{i_1 i_2\cdots i_N j_1 j_2\cdots j_M} = a_{i_1 i_2\cdots i_N} b_{j_1 j_2\cdots j_M}$
$\mathcal{X} = \mathbf{a}^{(1)}\circ\mathbf{a}^{(2)}\circ\cdots\circ\mathbf{a}^{(N)}$	TENSOR OR OUTER PRODUCT OF VECTORS $\mathbf{a}^{(n)}\in\mathbb{R}^{I_n}$ $(n=1,\ldots,N)$ YIELDS A RANK-1 TENSOR $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ WITH ENTRIES $x_{i_1 i_2\ldots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2}\ldots a^{(N)}_{i_N}$
$\mathbf{C}=\mathbf{A}\otimes\mathbf{B}$	KRONECKER PRODUCT OF $\mathbf{A}\in\mathbb{R}^{I_1\times I_2}$ AND $\mathbf{B}\in\mathbb{R}^{J_1\times J_2}$ YIELDS $\mathbf{C}\in\mathbb{R}^{I_1J_1\times I_2J_2}$ WITH ENTRIES $c_{(i_1-1)J_1+j_1,(i_2-1)J_2+j_2} = a_{i_1i_2}\, b_{j_1j_2}$
$\mathbf{C}=\mathbf{A}\odot\mathbf{B}$	KHATRI–RAO PRODUCT OF $\mathbf{A}=[\mathbf{a}_1,\ldots,\mathbf{a}_R]\in\mathbb{R}^{I\times R}$ AND $\mathbf{B}=[\mathbf{b}_1,\ldots,\mathbf{b}_R]\in\mathbb{R}^{J\times R}$ YIELDS $\mathbf{C}\in\mathbb{R}^{IJ\times R}$ WITH COLUMNS $\mathbf{c}_r = \mathbf{a}_r\otimes\mathbf{b}_r$



[FIG2] Construction of tensors. (a) The tensorization of a vector or matrix into the so-called quantized format; in scientific computing, this facilitates supercompression of large-scale vectors or matrices. (b) The tensor is formed through the discretization of a trivariate function $f(x,y,z)$.

cases of (1), and owe their uniqueness to hard and restrictive constraints such as triangularity and orthogonality. On the other hand, certain properties of the factors in (1) can be represented by appropriate constraints, making possible the unique estimation or extraction of such factors. These constraints include statistical independence, sparsity, nonnegativity, exponential structure, uncorrelatedness, constant modulus, finite alphabet, smoothness, and unimodality. Indeed, the first four properties form the basis of ICA [12]–[14], SCA [32], NMF [21], and harmonic retrieval [35].

TENSORIZATION—BLESSING OF DIMENSIONALITY

While one-way (vectors) and two-way (matrices) algebraic structures were, respectively, introduced as natural representations for segments of scalar measurements and measurements on a grid, tensors were initially used purely for the mathematical benefits they provide in data analysis; for instance, it seemed natural to stack together excitation–emission spectroscopy matrices in chemometrics into a third-order tensor [7].

The procedure of creating a data tensor from lower-dimensional original data is referred to as *tensorization*, and we propose the following taxonomy for tensor generation:

1) *Rearrangement of lower-dimensional data structures*: Large-scale vectors or matrices are readily tensorized to higher-order tensors and can be compressed through tensor decompositions if they admit a low-rank tensor approximation; this principle facilitates big data analysis [23], [29], [30] [see Figure 2(a)]. For instance, a one-way exponential signal $x(k) = az^k$ can be rearranged into a rank-1 Hankel matrix or a Hankel tensor [36]

$$\mathbf{H} = \begin{pmatrix} x(0) & x(1) & x(2) & \cdots \\ x(1) & x(2) & x(3) & \cdots \\ x(2) & x(3) & x(4) & \cdots \\ \vdots & \vdots & \vdots & \end{pmatrix} = a\,\boldsymbol{b} \circ \boldsymbol{b}, \tag{2}$$

where $\boldsymbol{b} = [1, z, z^2, \ldots]^T$. Also, in sensor array processing, tensor structures naturally emerge when combining snapshots from identical subarrays [19].

2) *Mathematical construction*: Among many such examples, the Nth-order moments (cumulants) of a vector-valued random variable form an Nth-order tensor [9], while in second-order ICA, snapshots of data statistics (covariance matrices) are effectively slices of a third-order tensor [12], [37]. Also, a (channel × time) data matrix can be transformed into a (channel × time × frequency) or (channel × time × scale) tensor via time-frequency or wavelet representations, a powerful procedure in multichannel electroencephalogram (EEG) analysis in brain science [21], [38].

3) *Experiment design*: Multifaceted data can be naturally stacked into a tensor; for instance, in wireless communications the so-called signal diversity (temporal, spatial, spectral, etc.) corresponds to the order of the tensor [20]. In the same spirit, the standard eigenfaces can be generalized to tensor faces by combining images with different illuminations, poses, and expressions [39], while the common modes in EEG recordings across subjects, trials, and conditions are best analyzed when combined together into a tensor [28].

4) *Natural tensor data*: Some data sources are readily generated as tensors [e.g., RGB color images, videos, three-dimensional (3-D) light field displays] [40]. Also, in scientific computing, we often need to evaluate a discretized multivariate function; this is a natural tensor, as illustrated in Figure 2(b) for a trivariate function $f(x, y, z)$ [23], [29], [30].

The high dimensionality of the tensor format is therefore associated with blessings, which include the possibilities to obtain compact representations, the uniqueness of decompositions, the flexibility in the choice of constraints, and the generality of components that can be identified.

CANONICAL POLYADIC DECOMPOSITION

DEFINITION

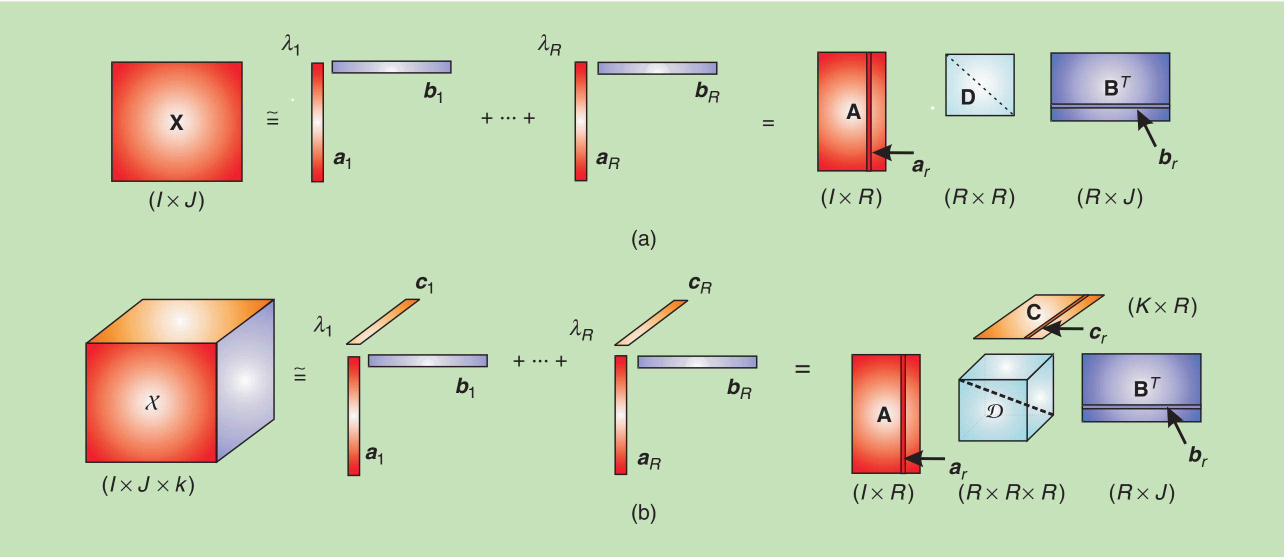
A polyadic decomposition (PD) represents an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ as a linear combination of rank-1 tensors in the form

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \boldsymbol{b}_r^{(1)} \circ \boldsymbol{b}_r^{(2)} \circ \cdots \circ \boldsymbol{b}_r^{(N)}. \tag{3}$$

Equivalently, $\mathcal{X}$ is expressed as a multilinear product with a diagonal core

$$\begin{aligned} \mathcal{X} &= \mathcal{D} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)} \\ &= [\![\mathcal{D}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!], \end{aligned} \tag{4}$$

where $\mathcal{D} = \mathrm{diag}_N(\lambda_1, \lambda_2, \ldots, \lambda_R)$ [cf. the matrix case in (1)]. Figure 3 illustrates these two interpretations for a third-order



[FIG3] **The analogy between (a) dyadic decompositions and (b) PDs; the Tucker format has a diagonal core. The uniqueness of these decompositions is a prerequisite for BSS and latent variable analysis.**

tensor. The tensor rank is defined as the smallest value of R for which (3) holds exactly; the minimum rank PD is called *canonical PD* (*CPD*) and is desired in signal separation. The term *CPD* may also be considered as an abbreviation of CANDECOMP/PARAFAC decomposition, see the "Historical Notes" section. The matrix/vector form of CPD can be obtained via the Khatri–Rao products (see Table 2) as

$$\mathbf{X}_{(n)} = \mathbf{B}^{(n)}\mathbf{D}(\mathbf{B}^{(N)} \odot \cdots \odot \mathbf{B}^{(n+1)} \odot \mathbf{B}^{(n-1)} \odot \cdots \odot \mathbf{B}^{(1)})^T,$$
$$\text{vec}(\mathcal{X}) = [\mathbf{B}^{(N)} \odot \mathbf{B}^{(N-1)} \odot \cdots \odot \mathbf{B}^{(1)}]\boldsymbol{d}, \quad (5)$$

where $\boldsymbol{d} = [\lambda_1, \lambda_2, \ldots, \lambda_R]^T$.

RANK

As mentioned earlier, the rank-related properties are very different for matrices and tensors. For instance, the number of complex-valued rank-1 terms needed to represent a higher-order tensor can be strictly smaller than the number of real-valued rank-1 terms [22], while the determination of tensor rank is in general NP-hard [41]. Fortunately, in signal processing applications, rank estimation most often corresponds to determining the number of tensor components that can be retrieved with sufficient accuracy, and often there are only a few data components present. A pragmatic first assessment of the number of components may be through inspection of the multilinear singular value spectrum (see the "Tucker Decomposition" section), which indicates the size of the core tensor in the right-hand side of Figure 3(b). The existing techniques for rank estimation include the core consistency diagnostic (CORCONDIA) algorithm, which checks whether the core tensor is (approximately) diagonalizable [7], while a number of techniques operate by balancing the approximation error versus the number of degrees of freedom for a varying number of rank-1 terms [42]–[44].

UNIQUENESS

Uniqueness conditions give theoretical bounds for exact tensor decompositions. A classical uniqueness condition is due to Kruskal [33], which states that for third-order tensors, the CPD is unique up to unavoidable scaling and permutation ambiguities, provided that $k_{\mathbf{B}^{(1)}} + k_{\mathbf{B}^{(2)}} + k_{\mathbf{B}^{(3)}} \geq 2R + 2$, where the Kruskal rank $k_{\mathbf{B}}$ of a matrix $\mathbf{B}$ is the maximum value ensuring that any subset of $k_{\mathbf{B}}$ columns is linearly independent. In sparse modeling, the term $(k_{\mathbf{B}} + 1)$ is also known as the spark [32]. A generalization to Nth-order tensors is due to Sidiropoulos and Bro [45] and is given by

$$\sum_{n=1}^{N} k_{\mathbf{B}^{(n)}} \geq 2R + N - 1. \quad (6)$$

More relaxed uniqueness conditions can be obtained when one factor matrix has full-column rank [46]–[48]; for a thorough study of the third-order case, we refer to [34]. This all shows that, compared to matrix decompositions, CPD is unique under more natural and relaxed conditions, which only require the components to be sufficiently different and their number not unreasonably large. These conditions do not have a matrix counterpart and are at the heart of tensor-based signal separation.

COMPUTATION

Certain conditions, including Kruskal's, enable explicit computation of the factor matrices in (3) using linear algebra [essentially, by solving sets of linear equations and computing (generalized) EVD] [6], [47], [49], [50]. The presence of noise in data means that CPD is rarely exact, and we need to fit a CPD model to the data by minimizing a suitable cost function. This is typically achieved by minimizing the Frobenius norm of the difference between the given data tensor and its CP approximation, or, alternatively, by least absolute error fitting when the noise is Laplacian [51]. The theoretical Cramér–Rao lower bound and

Cramér–Rao induced bound for the assessment of CPD performance were derived in [52] and [53].

Since the computation of CPD is intrinsically multilinear, we can arrive at the solution through a sequence of linear subproblems as in the alternating least squares (ALS) framework, whereby the least squares (LS) cost function is optimized for one component matrix at a time, while keeping the other component matrices fixed [6]. As seen from (5), such a conditional update scheme boils down to solving overdetermined sets of linear equations.

While the ALS is attractive for its simplicity and satisfactory performance for a few well-separated components and at sufficiently high signal-to-noise ratio (SNR), it also inherits the problems of alternating algorithms and is not guaranteed to converge to a stationary point. This can be rectified by only updating the factor matrix for which the cost function has most decreased at a given step [54], but this results in an N-times increase in computational cost per iteration. The convergence of ALS is not yet completely understood—it is quasilinear close to the stationary point [55], while it becomes rather slow for ill-conditioned cases; for more details, we refer to [56] and [57].

The conventional all-at-once algorithms for numerical optimization, such as nonlinear conjugate gradients, quasi-Newton, or nonlinear least squares (NLS) [58], [59], have been shown to often outperform ALS for ill-conditioned cases and to be typically more robust to overfactoring. However, these come at the cost of a much higher computational load per iteration. More sophisticated versions use the rank-1 structure of the terms within CPD to perform efficient computation and storage of the Jacobian and (approximate) Hessian; their complexity is on par with ALS while, for ill-conditioned cases, the performance is often superior [60], [61].

An important difference between matrices and tensors is that the existence of a best rank-R approximation of a tensor of rank greater than R is not guaranteed [22], [62] since the set of tensors whose rank is at most R is not closed. As a result, the cost functions for computing factor matrices may only have an infimum (instead of a minimum) so that their minimization will approach the boundary of that set without ever reaching the boundary point. This will cause two or more rank-1 terms go to infinity upon convergence of an algorithm; however, numerically, the diverging terms will almost completely cancel one another while the overall cost function will still decrease along the iterations [63]. These diverging terms indicate an inappropriate data model: the mismatch between the CPD and the original data tensor may arise because of an underestimated number of components, not all tensor components having a rank-1 structure, or data being too noisy.

CONSTRAINTS

As mentioned earlier, under quite mild conditions, the CPD is unique by itself, without requiring additional constraints. However, to enhance the accuracy and robustness with respect to noise, prior knowledge of data properties (e.g., statistical independence, sparsity) may be incorporated into the constraints on factors so as to facilitate their physical interpretation, relax the uniqueness conditions, and even simplify computation [64]–[66]. Moreover, the orthogonality and nonnegativity constraints ensure the existence of the minimum of the optimization criterion used [63], [64], [67].

APPLICATIONS

The CPD has already been established as an advanced tool for signal separation in vastly diverse branches of signal processing and data analysis, such as in audio and speech processing, biomedical engineering, chemometrics, and machine learning [7], [24], [25], [28]. Note that algebraic ICA algorithms are effectively based on the CPD of a tensor of the statistics of recordings; the statistical independence of the sources is reflected in the diagonality of the core tensor in Figure 3, i.e., in vanishing cross-statistics [11], [12]. The CPD is also heavily used in exploratory data analysis, where the rank-1 terms capture the essential properties of dynamically complex signals [8]. Another example is in wireless communication, where the signals transmitted by different users correspond to rank-1 terms in the case of line-of-sight propagation [19]. Also, in harmonic retrieval and direction of arrival type applications, real or complex exponentials have a rank-1 structure, for which the use of CPD is natural [36], [65].

EXAMPLE 1

Consider a sensor array consisting of K displaced but otherwise identical subarrays of I sensors, with $\tilde{I} = KI$ sensors in total. For R narrowband sources in the far field, the baseband equivalent model of the array output becomes $\mathbf{X} = \mathbf{A}\mathbf{S}^T + \mathbf{E}$, where $\mathbf{A} \in \mathbb{C}^{\tilde{I} \times R}$ is the global array response, $\mathbf{S} \in \mathbb{C}^{J \times R}$ contains J snapshots of the sources, and $\mathbf{E}$ is the noise. A single source $(R = 1)$ can be obtained from the best rank-1 approximation of the matrix $\mathbf{X}$; however, for $R > 1$, the decomposition of $\mathbf{X}$ is not unique, and, hence, the separation of sources is not possible without incorporating additional information. The constraints on the sources that may yield a unique solution are, for instance, constant modulus and statistical independence [12], [68].

Consider a row-selection matrix $\mathbf{J}_k \in \mathbb{C}^{I \times \tilde{I}}$ that extracts the rows of $\mathbf{X}$ corresponding to the kth subarray, $k = 1, \ldots, K$. For two identical subarrays, the generalized EVD of the matrices $\mathbf{J}_1\mathbf{X}$ and $\mathbf{J}_2\mathbf{X}$ corresponds to the well-known estimation of signal parameters via rotational invariance techniques (ESPRIT) [69]. For the case $K > 2$, we shall consider $\mathbf{J}_k\mathbf{X}$ as slices of the tensor $\mathcal{X} \in \mathbb{C}^{I \times J \times K}$ (see the section "Tensorization—Blessing of Dimensionality"). It can be shown that the signal part of $\mathcal{X}$ admits a CPD as in (3) and (4), with $\lambda_1 = \cdots = \lambda_R = 1$, $\mathbf{J}_k\mathbf{A} = \mathbf{B}^{(1)} \operatorname{diag}(b_{k1}^{(3)}, \ldots, b_{kR}^{(3)})$, and $\mathbf{B}^{(2)} = \mathbf{S}$ [19], and the consequent source separation under rather mild conditions—its uniqueness does not require constraints such as statistical independence or constant modulus. Moreover, the decomposition is unique even in cases when the number of sources, R, exceeds the number of subarray sensors, I, or even the total number of sensors, $\tilde{I}$. Note that particular array geometries, such as linearly and uniformly displaced subarrays, can be converted into a constraint on CPD, yielding a further relaxation of the uniqueness conditions, reduced sensitivity to noise, and often faster computation [65].

TUCKER DECOMPOSITION

Figure 4 illustrates the principle of TKD, which treats a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ as a multilinear transformation of a (typically dense but small) core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ by the factor matrices $\mathbf{B}^{(n)} = [\boldsymbol{b}_1^{(n)}, \boldsymbol{b}_2^{(n)}, \ldots, \boldsymbol{b}_{R_n}^{(n)}] \in \mathbb{R}^{I_n \times R_n}$, $n = 1, 2, \ldots, N$ [3], [4], given by

$$\mathcal{X} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \cdots r_N} (\boldsymbol{b}_{r_1}^{(1)} \circ \boldsymbol{b}_{r_2}^{(2)} \circ \cdots \circ \boldsymbol{b}_{r_N}^{(N)}), \tag{7}$$

or equivalently

$$\begin{aligned} \mathcal{X} &= \mathcal{G} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)} \\ &= [\![\mathcal{G}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!]. \end{aligned} \tag{8}$$

Via the Kronecker products (see Table 2), TKD can be expressed in a matrix/vector form as

$$\begin{aligned} \mathbf{X}_{(n)} &= \mathbf{B}^{(n)} \mathbf{G}_{(n)} (\mathbf{B}^{(N)} \otimes \cdots \otimes \mathbf{B}^{(n+1)} \otimes \mathbf{B}^{(n-1)} \otimes \cdots \otimes \mathbf{B}^{(1)})^T \\ \operatorname{vec}(\mathcal{X}) &= [\mathbf{B}^{(N)} \otimes \mathbf{B}^{(N-1)} \otimes \cdots \otimes \mathbf{B}^{(1)}] \operatorname{vec}(\mathcal{G}). \end{aligned}$$

Although Tucker initially used the orthogonality and ordering constraints on the core tensor and factor matrices [3], [4], we can also employ other meaningful constraints.

MULTILINEAR RANK

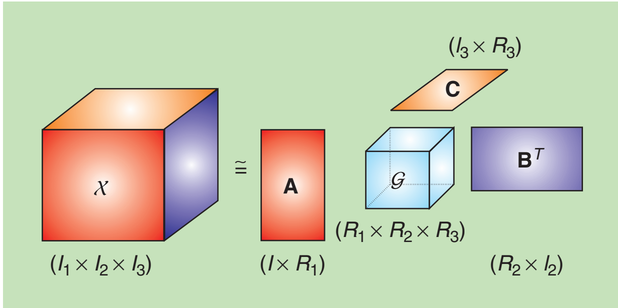
For a core tensor of minimal size, R_1 is the column rank (the dimension of the subspace spanned by mode-1 fibers), R_2 is the row rank (the dimension of the subspace spanned by mode-2 fibers), and so on. A remarkable difference from matrices is that the values of $R_1, R_2, \ldots, R_N$ can be different for $N \geq 3$. The N-tuple $(R_1, R_2, \ldots, R_N)$ is consequently called the *multilinear rank* of the tensor $\mathcal{X}$.

LINKS BETWEEN CPD AND TUCKER DECOMPOSTION

TKD can be considered an expansion in rank-1 terms (polyadic but not necessary canonical), as shown in (7), while (4) represents CPD as a multilinear product of a core tensor and factor matrices (but the core is not necessary minimal); Table 3 shows various other connections. However, despite the obvious interchangeability of notation, the CPD and TKD serve different purposes. In general, the Tucker core cannot be diagonalized, while the number of CPD terms may not be bounded by the multilinear rank. Consequently, in signal processing and data analysis, CPD is typically used for factorizing data into easy to interpret components (i.e., the rank-1 terms), while the goal of unconstrained TKD is most often to compress data into a tensor of smaller size (i.e., the core tensor) or to find the subspaces spanned by the fibers (i.e., the column spaces of the factor matrices).

UNIQUENESS

The unconstrained TKD is in general not unique, i.e., factor matrices $\mathbf{B}^{(n)}$ are rotation invariant. However, physically, the subspaces defined by the factor matrices in TKD are unique, while the bases in these subspaces may be chosen arbitrarily—their choice is compensated for within the core tensor. This becomes clear upon



[FIG4] The Tucker decompostion of a third-order tensor. The column spaces of A, B, and C represent the signal subspaces for the three modes. The core tensor $\mathcal{G}$ is nondiagonal, accounting for the possibly complex interactions among tensor components.

realizing that any factor matrix in (8) can be postmultiplied by any nonsingular (rotation) matrix; in turn, this multiplies the core tensor by its inverse, i.e.,

$$\begin{aligned} \mathcal{X} &= [\![\mathcal{G}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!] \\ &= [\![\mathcal{H}; \mathbf{B}^{(1)}\mathbf{R}^{(1)}, \mathbf{B}^{(2)}\mathbf{R}^{(2)}, \ldots, \mathbf{B}^{(N)}\mathbf{R}^{(N)}]\!], \\ \mathcal{H} &= [\![\mathcal{G}; \mathbf{R}^{(1)^{-1}}, \mathbf{R}^{(2)^{-1}}, \ldots, \mathbf{R}^{(N)^{-1}}]\!], \end{aligned} \tag{9}$$

where the matrices $\mathbf{R}^{(n)}$ are invertible.

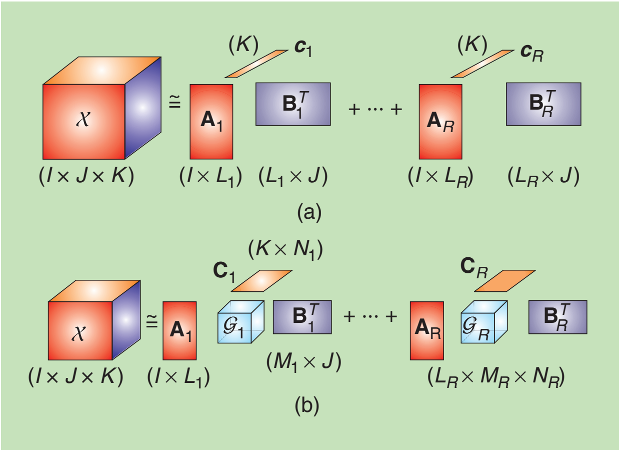
MULTILINEAR SVD

Orthonormal bases in a constrained Tucker representation can be obtained via the SVD of the mode-n matricized tensor $\mathbf{X}_{(n)} = \mathbf{U}_n \boldsymbol{\Sigma}_n \mathbf{V}_n^T$ (i.e., $\mathbf{B}^{(n)} = \mathbf{U}_n$, $n = 1, 2, \ldots, N$). Because of the orthonormality, the corresponding core tensor becomes

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_N \mathbf{U}_N^T. \tag{10}$$

[TABLE 3] DIFFERENT FORMS OF CPD AND TUCKER REPRESENTATIONS OF A THIRD-ORDER TENSOR $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$.

CPD	TKD
TENSOR REPRESENTATION, OUTER PRODUCTS	
$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$	$\mathcal{X} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} \boldsymbol{a}_{r_1} \circ \boldsymbol{b}_{r_2} \circ \boldsymbol{c}_{r_3}$
TENSOR REPRESENTATION, MULTILINEAR PRODUCTS	
$\mathcal{X} = \mathcal{D} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$	$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$
MATRIX REPRESENTATIONS	
$\mathbf{X}_{(1)} = \mathbf{A}\,\mathbf{D}(\mathbf{C} \odot \mathbf{B})^T$	$\mathbf{X}_{(1)} = \mathbf{A}\,\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^T$
$\mathbf{X}_{(2)} = \mathbf{B}\,\mathbf{D}(\mathbf{C} \odot \mathbf{A})^T$	$\mathbf{X}_{(2)} = \mathbf{B}\,\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^T$
$\mathbf{X}_{(3)} = \mathbf{C}\,\mathbf{D}(\mathbf{B} \odot \mathbf{A})^T$	$\mathbf{X}_{(3)} = \mathbf{C}\,\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^T$
VECTOR REPRESENTATION	
$\operatorname{vec}(\mathcal{X}) = (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})\boldsymbol{d}$	$\operatorname{vec}(\mathcal{X}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A}) \operatorname{vec}(\mathcal{G})$
SCALAR REPRESENTATION	
$x_{ijk} = \sum_{r=1}^{R} \lambda_r a_{ir} b_{jr} c_{kr}$	$x_{ijk} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} a_{ir_1} b_{jr_2} c_{kr_3}$
MATRIX SLICES $\mathbf{X}_k = \mathbf{X}(:,:,k)$	
$\mathbf{X}_k = \mathbf{A} \operatorname{diag}(c_{k1}, c_{k2}, \ldots, c_{kR}) \mathbf{B}^T$	$\mathbf{X}_k = \mathbf{A} \sum_{r_3=1}^{R_3} c_{kr_3} \mathbf{G}(:,:,r_3) \mathbf{B}^T$



[FIG5] **BTDs find data components that are structurally more complex than the rank-1 terms in CPD. (a) Decomposition into terms with multilinear rank $(L_r, L_r, 1)$. (b) Decomposition into terms with multilinear rank (L_r, M_r, N_r).**

Then, the singular values of $\mathbf{X}_{(n)}$ are the Frobenius norms of the corresponding slices of the core tensor $\mathcal{S}$: $(\Sigma_n)_{r_n, r_n} = \| \mathcal{S}(:,:,\ldots,r_n,:,\ldots,:) \|_F$, with slices in the same mode being mutually orthogonal, i.e., their inner products are zero. The columns of $\mathbf{U}_n$ may thus be seen as multilinear singular vectors, while the norms of the slices of the core are multilinear singular values [15]. As in the matrix case, the multilinear singular values govern the multilinear rank, while the multilinear singular vectors allow, for each mode separately, an interpretation as in PCA [8].

LOW MULTILINEAR RANK APPROXIMATION

Analogous to PCA, a large-scale data tensor $\mathcal{X}$ can be approximated by discarding the multilinear singular vectors and slices of the core tensor that correspond to small multilinear singular values, i.e., through truncated matrix SVDs. Low multilinear rank approximation is always well posed; however, the truncation is not necessarily optimal in the LS sense, although a good estimate can often be made as the approximation error corresponds to the degree of truncation. When it comes to finding the best approximation, the ALS-type algorithms exhibit similar advantages and drawbacks to those used for CPD [8], [70]. Optimization-based algorithms exploiting second-order information have also been proposed [71], [72].

CONSTRAINTS AND TUCKER-BASED MULTIWAY COMPONENT ANALYSIS

Besides orthogonality, constraints that may help to find unique basis vectors in a Tucker representation include statistical independence, sparsity, smoothness, and nonnegativity [21], [73], [74]. Components of a data tensor seldom have the same properties in its modes, and for physically meaningful representation, different constraints may be required in different modes so as to match the properties of the data at hand. Figure 1 illustrates the concept of multiway component analysis (MWCA) and its flexibility in choosing the modewise constraints; a Tucker representation of MWCA naturally accommodates such diversities in different modes.

OTHER APPLICATIONS

We have shown that TKD may be considered a multilinear extension of PCA [8]; it therefore generalizes signal subspace techniques, with applications including classification, feature extraction, and subspace-based harmonic retrieval [27], [39], [75], [76]. For instance, a low multilinear rank approximation achieved through TKD may yield a higher SNR than the SNR in the original raw data tensor, making TKD a very natural tool for compression and signal enhancement [7], [8], [26].

BLOCK TERM DECOMPOSITIONS

We have already shown that CPD is unique under quite mild conditions. A further advantage of tensors over matrices is that it is even possible to relax the rank-1 constraint on the terms, thus opening completely new possibilities in, e.g., BSS. For clarity, we shall consider the third-order case, whereby, by replacing the rank-1 matrices $\boldsymbol{b}_r^{(1)} \circ \boldsymbol{b}_r^{(2)} = \boldsymbol{b}_r^{(1)} \boldsymbol{b}_r^{(2)T}$ in (3) by low-rank matrices $\mathbf{A}_r \mathbf{B}_r^T$, the tensor $\mathcal{X}$ can be represented as [Figure 5(a)]

$$\mathcal{X} = \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^T) \circ \boldsymbol{c}_r. \tag{11}$$

Figure 5(b) shows that we can even use terms that are only required to have a low multilinear rank (see the "Tucker Decomposition" section) to give

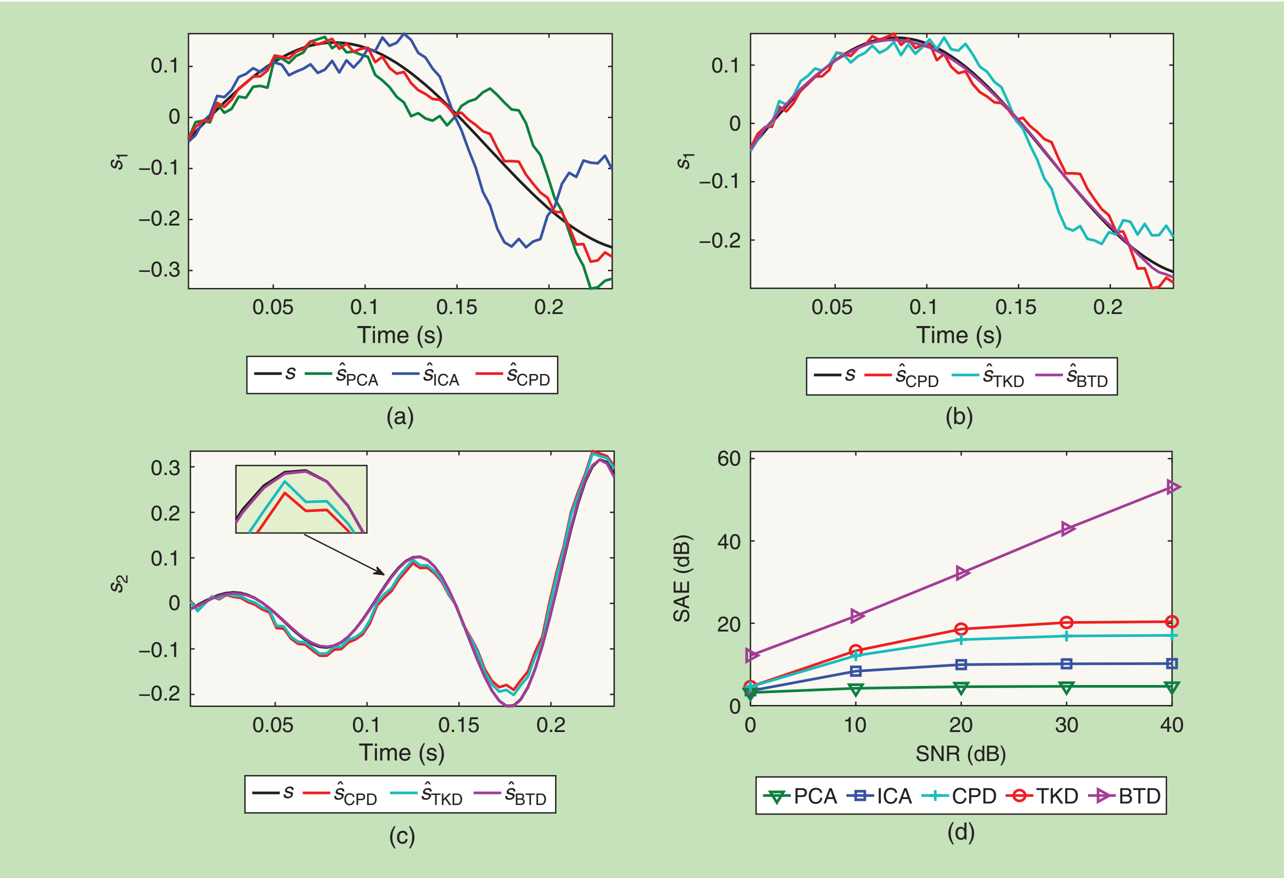
$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{G}_r \times_1 \mathbf{A}_r \times_2 \mathbf{B}_r \times_3 \mathbf{C}_r. \tag{12}$$

These so-called block term decompositions (BTDs) in (11) and (12) admit the modeling of more complex signal components than CPD and are unique under more restrictive but still fairly natural conditions [77]–[79].

EXAMPLE 2

To compare some standard and tensor approaches for the separation of short duration correlated sources, BSS was performed on five linear mixtures of the sources $s_1(t) = \sin(6\pi t)$ and $s_2(t) = \exp(10t)\sin(20\pi t)$, which were contaminated by white Gaussian noise, to give the mixtures $\mathbf{X} = \mathbf{AS} + \mathbf{E} \in \mathbb{R}^{5 \times 60}$, where $\mathbf{S}(t) = [\boldsymbol{s}_1(t), \boldsymbol{s}_2(t)]^T$ and $\mathbf{A} \in \mathbb{R}^{5 \times 2}$ was a random matrix whose columns (mixing vectors) satisfy $\boldsymbol{a}_1^T \boldsymbol{a}_2 = 0.1$, $\|\boldsymbol{a}_1\|_2 = \|\boldsymbol{a}_2\|_2 = 1$. The 3-Hz sine wave did not complete a full period over the 60 samples so that the two sources had a correlation degree of $(|\boldsymbol{s}_1^T \boldsymbol{s}_2|)/(\|\boldsymbol{s}_1\|_2 \|\boldsymbol{s}_2\|_2) = 0.35$. The tensor approaches, CPD, TKD, and BTD employed a third-order tensor $\mathcal{X}$ of size 24 × 37 × 5 generated from five Hankel matrices whose elements obey $\mathcal{X}(i, j, k) = \mathbf{X}(k, i + j - 1)$ (see the section "Tensorization—Blessing of Dimensionality"). The average squared angular error (SAE) was used as the performance measure. Figure 6 shows the simulation results, illustrating the following.

- PCA failed since the mixing vectors were not orthogonal and the source signals were correlated, both violating the assumptions for PCA.
- The ICA [using the joint approximate diagonalization of eigenmatrices (JADE) algorithm [10]] failed because the signals were not statistically independent, as assumed in ICA.



[FIG6] The blind separation of the mixture of a pure sine wave and an exponentially modulated sine wave using PCA, ICA, CPD, TKD, and BTD. The sources s_1 and s_2 are correlated and of short duration; the symbols $\hat{s}_1$ and $\hat{s}_2$ denote the estimated sources. (a)–(c) Sources $s_1(t)$ and $s_2(t)$ and their estimates using PCA, ICA, CPD, TKD, and BTD; (d) average squared angular errors (SAE) in estimation of the sources.

- Low-rank tensor approximation via a rank-2 CPD was used to estimate $\mathbf{A}$ as the third factor matrix, which was then inverted to yield the sources. The accuracy of CPD was compromised as the components of tensor $\mathcal{X}$ cannot be represented by rank-1 terms.
- Low multilinear rank approximation via TKD for the multilinear rank (4, 4, 2) was able to retrieve the column space of the mixing matrix but could not find the individual mixing vectors because of the nonuniqueness of TKD.
- BTD in multilinear rank-(2, 2, 1) terms matched the data structure [78]; it is remarkable that the sources were recovered using as few as six samples in the noise-free case.

HIGHER-ORDER COMPRESSED SENSING (HO-CS)

The aim of CS is to provide a faithful reconstruction of a signal of interest, even when the set of available measurements is (much) smaller than the size of the original signal [80]–[83]. Formally, we have available M (compressive) data samples $\boldsymbol{y} \in \mathbb{R}^M$, which are assumed to be linear transformations of the original signal $\boldsymbol{x} \in \mathbb{R}^I$ $(M < I)$. In other words, $\boldsymbol{y} = \boldsymbol{\Phi x}$, where the sensing matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times I}$ is usually random. Since the projections are of a lower dimension than the original data, the reconstruction is an ill-posed inverse problem whose solution requires knowledge of the physics of the problem converted into constraints. For example, a two-dimensional image $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ can be vectorized as a long vector $\boldsymbol{x} = \text{vec}(\mathbf{X}) \in \mathbb{R}^I$ $(I = I_1 I_2)$ that admits sparse representation in a known dictionary $\mathbf{B} \in \mathbb{R}^{I \times I}$ so that $\boldsymbol{x} = \mathbf{B}\boldsymbol{g}$, where the matrix $\mathbf{B}$ may be a wavelet or discrete cosine transform dictionary. Then, faithful recovery of the original signal $\boldsymbol{x}$ requires finding the sparsest vector $\boldsymbol{g}$ such that

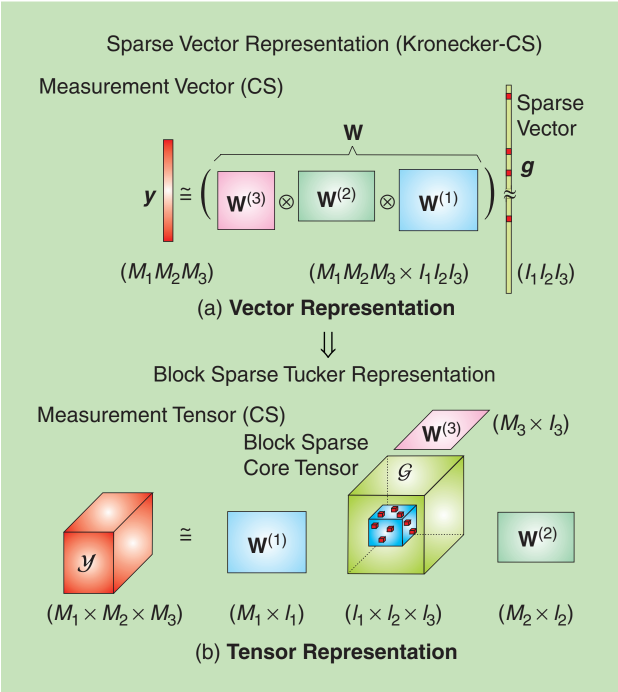
$$\boldsymbol{y} = \mathbf{W}\boldsymbol{g}, \text{ with } \|\boldsymbol{g}\|_0 \le K, \qquad \mathbf{W} = \boldsymbol{\Phi}\mathbf{B}, \tag{13}$$

where $\|\cdot\|_0$ is the ℓ_0-norm (number of nonzero entries) and $K \ll I$.

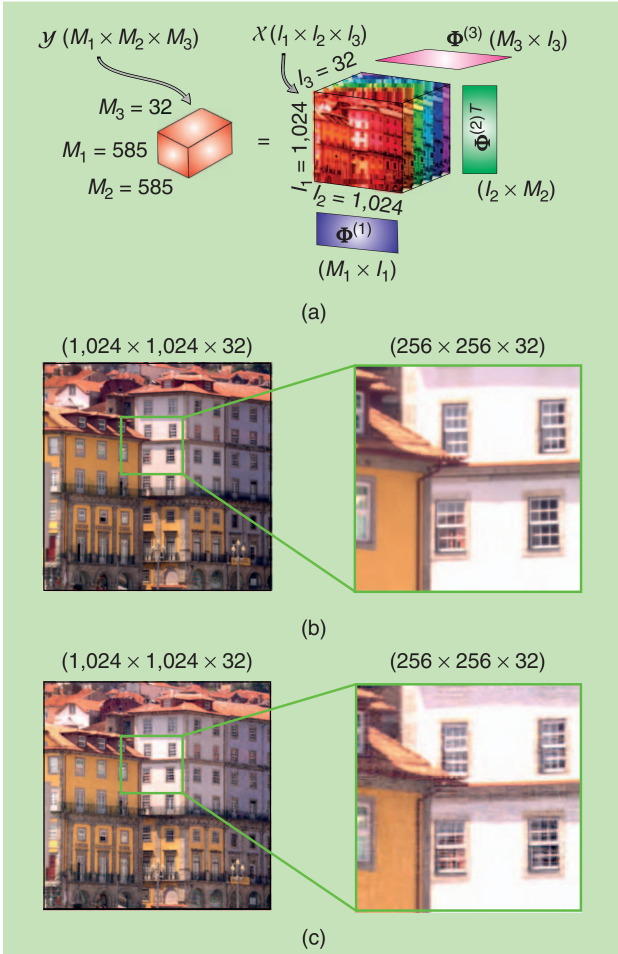
Since the ℓ_0-norm minimization is not practical, alternative solutions involve iterative refinements of the estimates of vector $\boldsymbol{g}$ using greedy algorithms such as the orthogonal matching pursuit (OMP) algorithm, or the ℓ_1-norm minimization algorithms $\left(\|\boldsymbol{g}\|_1 = \sum_{i=1}^{I} |g_i|\right)$ [83]. Low coherence of the composite dictionary matrix $\mathbf{W}$ is a prerequisite for a satisfactory recovery of $\boldsymbol{g}$ (and hence $\boldsymbol{x}$)—we need to choose $\boldsymbol{\Phi}$ and $\mathbf{B}$ so that the correlation between the columns of $\mathbf{W}$ is minimum [83].

When extending the CS framework to tensor data, we face two obstacles:

- loss of information, such as spatial and contextual relationships in data, when a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is vectorized.



[FIG7] CS with a Kronecker-structured dictionary. OMP can perform faster if the sparse entries belong to a small subtensor, up to permutation of the columns of $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{W}^{(3)}$.



[FIG8] The multidimensional CS of a 3-D hyperspectral image using Tucker representation with a small sparse core in wavelet bases. (a) The Kronecker-CS of a 32-channel hyperspectral image. (b) The original hyperspectral image-RGB display. (c) The reconstruction (SP = 33%, PSNR = 35.51 dB)-RGB display.

- data handling since the size of vectorized data and the associated dictionary $\mathbf{B} \in \mathbb{R}^{I \times I}$ easily becomes prohibitively large (see the section "Large-Scale Data and the Curse of Dimensionality"), especially for tensors of high order.

Fortunately, tensor data are typically highly structured, a perfect match for compressive sampling, so that the CS framework relaxes data acquisition requirements, enables compact storage, and facilitates data completion (i.e., inpainting of missing samples due to a faulty sensor or unreliable measurement).

KRONECKER-CS FOR FIXED DICTIONARIES

In many applications, the dictionary and the sensing matrix admit a Kronecker structure (Kronecker-CS model), as illustrated in Figure 7(a) [84]. In this way, the global composite dictionary matrix becomes $\mathbf{W} = \mathbf{W}^{(N)} \otimes \mathbf{W}^{(N-1)} \otimes \cdots \otimes \mathbf{W}^{(1)}$, where each term $\mathbf{W}^{(n)} = \boldsymbol{\Phi}^{(n)}\mathbf{B}^{(n)}$ has a reduced dimensionality since $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ and $\boldsymbol{\Phi}^{(n)} \in \mathbb{R}^{M_n \times I_n}$. Denote $M = M_1 M_2 \cdots M_N$ and $I = I_1 I_2 \cdots I_N$, then, since $M_n \le I_n$, $n = 1, 2, \ldots, N$, this reduces storage requirements by a factor of $(\Sigma_n I_n M_n)/(MI)$. The computation of $\mathbf{W}\boldsymbol{g}$ is affordable since $\boldsymbol{g}$ is sparse; however, computing $\mathbf{W}^T\boldsymbol{y}$ is expensive but can be efficiently implemented through a sequence of products involving much smaller matrices $\mathbf{W}^{(n)}$ [85]. We refer to [84] for links between the coherence of factor matrices $\mathbf{W}^{(n)}$ and the coherence of the global composite dictionary matrix $\mathbf{W}$.

Figure 7 and Table 3 illustrate that the Kronecker-CS model is effectively a vectorized TKD with a sparse core. The tensor equivalent of the CS paradigm in (13) is therefore to find the sparsest core tensor $\mathcal{G}$ such that

$$\mathcal{Y} \cong \mathcal{G} \times_1 \mathbf{W}^{(1)} \times_2 \mathbf{W}^{(2)} \cdots \times_N \mathbf{W}^{(N)}, \qquad (14)$$

with $\|\mathcal{G}\|_0 \le K$, for a given set of modewise dictionaries $\mathbf{B}^{(n)}$ and sensing matrices $\boldsymbol{\Phi}^{(n)}$ $(n = 1, 2, \ldots, N)$. Working with several small dictionary matrices, appearing in a Tucker representation, instead of a large global dictionary matrix, is an example of the use of tensor structure for efficient representation; see also the section "Large-Scale Data and the Curse of Dimensionality."

A higher-order extension of the OMP algorithm, referred to as the *Kronecker-OMP algorithm* [85], requires K iterations to find the K nonzero entries of the core tensor $\mathcal{G}$. Additional computational advantages can be gained if it can be assumed that the K nonzero entries belong to a small subtensor of $\mathcal{G}$, as shown in Figure 7(b); such a structure is inherent to, e.g., hyperspectral imaging [85], [86] and 3-D astrophysical signals. More precisely, if the $K = L^N$ nonzero entries are located within a subtensor of size $(L \times L \times \cdots \times L)$, where $L \ll I_n$, then, by exploiting the block-tensor structure, the so-called N-way block OMP algorithm (N-BOMP) requires at most NL iterations, which is linear in N

[85]. The Kronecker-CS model has been applied in magnetic resonance imaging, hyperspectral imaging, and in the inpainting of multiway data [86], [84].

APPROACHES WITHOUT FIXED DICTIONARIES

In Kronecker-CS, the modewise dictionaries $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ can be chosen so as best to represent the physical properties or prior knowledge about the data. They can also be learned from a large ensemble of data tensors, for instance, in an ALS-type fashion [86]. Instead of the total number of sparse entries in the core tensor, the size of the core (i.e., the multilinear rank) may be used as a measure for sparsity so as to obtain a low-complexity representation from compressively sampled data [87], [88]. Alternatively, a CPD representation can be used instead of a Tucker representation. Indeed, early work in chemometrics involved excitation–emission data for which part of the entries was unreliable because of scattering; the CPD of the data tensor is then computed by treating such entries as missing [7]. While CS variants of several CPD algorithms exist [59], [89], the oracle properties of tensor-based models are still not as well understood as for their standard models; a notable exception is CPD with sparse factors [90].

EXAMPLE 3

Figure 8 shows an original 3-D ($1{,}024 \times 1{,}024 \times 32$) hyperspectral image $\mathcal{X}$, which contains scene reflectance measured at 32 different frequency channels, acquired by a low-noise Peltier-cooled digital camera in the wavelength range of 400–720 nm [91]. Within the Kronecker-CS setting, the tensor of compressive measurements $\mathcal{Y}$ was obtained by multiplying the frontal slices by random Gaussian sensing matrices $\mathbf{\Phi}^{(1)} \in \mathbb{R}^{M_1 \times 1024}$ and $\mathbf{\Phi}^{(2)} \in \mathbb{R}^{M_2 \times 1024}$ $(M_1, M_2 < 1{,}024)$ in the first and second mode, respectively, while $\mathbf{\Phi}^{(3)} \in \mathbb{R}^{32 \times 32}$ was the identity matrix [see Figure 8(a)]. We used Daubechies wavelet factor matrices $\mathbf{B}^{(1)} = \mathbf{B}^{(2)} \in \mathbb{R}^{1024 \times 1024}$ and $\mathbf{B}^{(3)} \in \mathbb{R}^{32 \times 32}$, and employed the N-way block tensor N-BOMP to recover the small sparse core tensor and, subsequently, reconstruct the original 3-D image, as shown in Figure 8(b). For the sampling ratio SP=33% $(M_1 = M_2 = 585)$ this gave the peak SNR (PSNR) of 35.51 dB, while taking 71 min for $N_{\text{iter}} = 841$ iterations needed to detect the subtensor which contains the most significant entries. For the same quality of reconstruction (PSNR=35.51 dB), the more conventional Kronecker-OMP algorithm found 0.1% of the wavelet coefficients as significant, thus requiring $N_{\text{iter}} = K = 0.001 \times (1{,}024 \times 1{,}024 \times 32) = 33{,}555$ iterations and days of computation time.

LARGE-SCALE DATA AND THE CURSE OF DIMENSIONALITY

The sheer size of tensor data easily exceeds the memory or saturates the processing capability of standard computers; it is, therefore, natural to ask ourselves how tensor decompositions can be computed if the tensor dimensions in all or some modes are large or, worse still, if the tensor order is high. The term *curse of dimensionality*, in a general sense, was introduced by Bellman to refer to various computational bottlenecks when dealing with high-dimensional settings. In the context of tensors, the curse of dimensionality refers to the fact that the number of elements of an Nth-order $(I \times I \times \cdots \times I)$ tensor, I^N, scales exponentially with the tensor order N. For example, the number of values of a discretized function in Figure 2(b) quickly becomes unmanageable in terms of both computations and storing as N increases. In addition to their standard use (signal separation, enhancement, etc.), tensor decompositions may be elegantly employed in this context as efficient representation tools. The first question is, which type of tensor decomposition is appropriate?

[TABLE 4] STORAGE COST OF TENSOR MODELS FOR AN Nth-ORDER TENSOR $\mathcal{X} \in \mathbb{R}^{I \times I \times \cdots \times I}$ FOR WHICH THE STORAGE REQUIREMENT FOR RAW DATA IS $\mathcal{O}(I^N)$.

1) CANONICAL POLYADIC DECOMPOSITION	$\mathcal{O}(NIR)$
2) TUCKER	$\mathcal{O}(NIR + R^N)$
3) TENSOR TRAIN	$\mathcal{O}(NIR^2)$
4) QUANTIZED TENSOR TRAIN	$\mathcal{O}(NR^2 \log_2(I))$

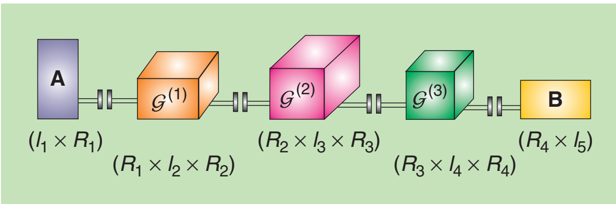
EFFICIENT DATA HANDLING

If all computations are performed on a CP representation and not on the raw data tensor itself, then, instead of the original I^N raw data entries, the number of parameters in a CP representation reduces to NIR, which scales linearly in N (see Table 4). This effectively bypasses the curse of dimensionality, while giving us the freedom to choose the rank, R, as a function of the desired accuracy [16]; on the other hand, the CP approximation may involve numerical problems (see the section "Canonical Polyadic Decomposition").

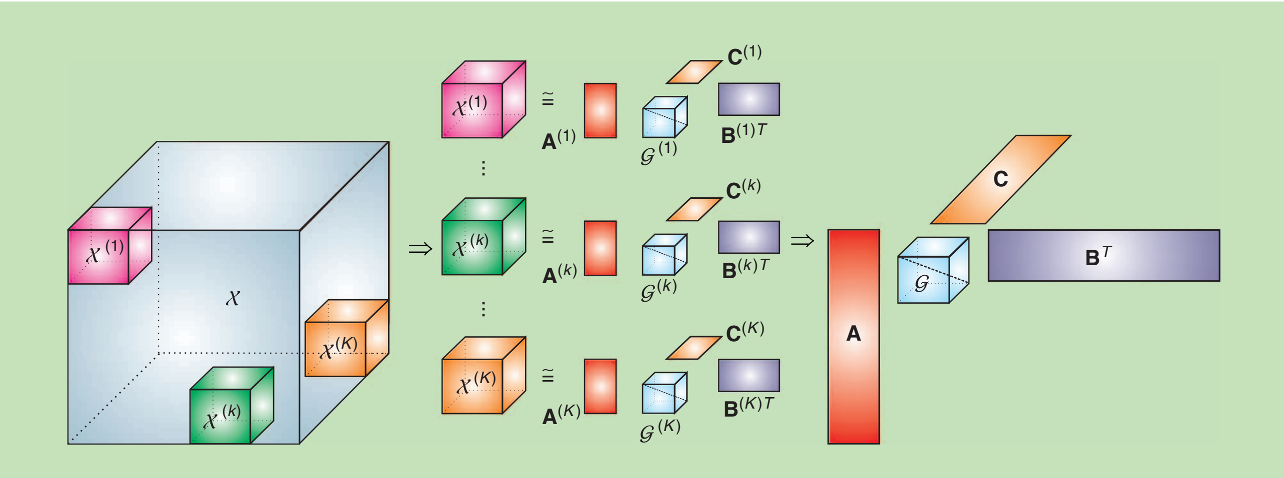
Compression is also inherent to TKD as it reduces the size of a given data tensor from the original I^N to $(NIR + R^N)$, thus exhibiting an approximate compression ratio of $(I/R)^N$. We can then benefit from the well understood and reliable approximation by means of matrix SVD; however, this is only useful for low N.

TENSOR NETWORKS

A numerically reliable way to tackle curse of dimensionality is through a concept from scientific computing and quantum information theory, termed *tensor networks*, which represents a tensor of a possibly very high order as a set of sparsely interconnected matrices and core tensors of low order (typically, order 3). These low-dimensional cores are interconnected via tensor contractions to provide a highly compressed representation of a data tensor. In addition, existing algorithms for the approximation of a given tensor by a tensor network have good numerical properties, making it



[FIG9] The TT decomposition of a fifth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_5}$, consisting of two matrix carriages and three third-order tensor carriages. The five carriages are connected through tensor contractions, which can be expressed in a scalar form as $x_{i_1,i_2,i_3,i_4,i_5} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_5=1}^{R_5} a_{i_1,r_1} g^{(1)}_{r_1,i_2,r_2} g^{(2)}_{r_2,i_3,r_3} g^{(3)}_{r_3,i_4,r_5} b_{r_4,i_5}$.



[FIG10] Efficient computation of CPD and TKD, whereby tensor decompositions are computed in parallel for sampled blocks. These are then merged to obtain the global components A, B, and C, and a core tensor $\mathcal{G}$.

possible to control the error and achieve any desired accuracy of approximation. For example, tensor networks allow for the representation of a wide class of discretized multivariate functions even in cases where the number of function values is larger than the number of atoms in the universe [23], [29], [30].

Examples of tensor networks are the hierarchical TKD and tensor trains (TTs) (see Figure 9) [17], [18]. The TTs are also known as matrix product states and have been used by physicists for more than two decades (see [92] and [93] and references therein). The PARATREE algorithm was developed in signal processing and follows a similar idea; it uses a polyadic representation of a data tensor (in a possibly nonminimal number of terms), whose computation then requires only the matrix SVD [94].

For very large-scale data that exhibit a well-defined structure, an even more radical approach to achieve a parsimonious representation may be through the concept of quantized or quantic tensor networks (QTNs) [29], [30]. For example, a huge vector $\mathbf{x} \in \mathbb{R}^I$ with $I = 2^L$ elements can be quantized and tensorized into a $(2 \times 2 \times \cdots \times 2)$ tensor $\mathcal{X}$ of order L, as illustrated in Figure 2(a). If $\mathbf{x}$ is an exponential signal, $x(k) = az^k$, then $\mathcal{X}$ is a symmetric rank-1 tensor that can be represented by two parameters: the scaling factor a and the generator z (cf. (2) in the section "Tensorization—Blessing of Dimensionality"). Nonsymmetric terms provide further opportunities, beyond the sum-of-exponential representation by symmetric low-rank tensors. Huge matrices and tensors may be dealt with in the same manner. For instance, an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, with $I_n = q^{L_n}$, can be quantized in all modes simultaneously to yield a $(q \times q \times \cdots \times q)$ quantized tensor of higher order. In QTN, q is small, typically $q = 2,3,4$, e.g., the binary encoding $(q = 2)$ reshapes an Nth -order tensor with $(2^{L_1} \times 2^{L_2} \times \cdots \times 2^{L_N})$ elements into a tensor of order $(L_1 + L_2 + \cdots + L_N)$ with the same number of elements. The TT decomposition applied to quantized tensors is referred to as the *quantized TT* (*QTT*); variants for other tensor representations have also been derived [29], [30]. In scientific computing, such formats provide the so-called supercompression—a logarithmic reduction of storage requirements: $\mathcal{O}(I^N) \rightarrow \mathcal{O}(N \log_q(I))$.

COMPUTATION OF THE DECOMPOSITION/REPRESENTATION

Now that we have addressed the possibilities for efficient tensor representation, the question that needs to be answered is how these representations can be computed from the data in an efficient manner. The first approach is to process the data in smaller blocks rather than in a batch manner [95]. In such a divide-and-conquer approach, different blocks may be processed in parallel, and their decompositions may be carefully recombined (see Figure 10) [95], [96]. In fact, we may even compute the decomposition through recursive updating as new data arrive [97]. Such recursive techniques may be used for efficient computation and for tracking decompositions in the case of nonstationary data.

The second approach would be to employ CS ideas (see the section "Higher-Order Compressed Sensing (HO-CS)") to fit an algebraic model with a limited number of parameters to possibly large data. In addition to enabling data completion (interpolation of missing data), this also provides a significant reduction of the cost of data acquisition, manipulation, and storage, breaking the curse of dimensionality being an extreme case.

While algorithms for this purpose are available both for low-rank and low multilinear rank representation [59], [87], an even more drastic approach would be to directly adopt sampled fibers as the bases in a tensor representation. In the TKD setting, we would choose the columns of the factor matrices $\mathbf{B}^{(n)}$ as mode-n fibers of the tensor, which requires us to address the following two problems: 1) how to find fibers that allow us to accurately represent the tensor and 2) how to compute the corresponding core tensor at a low cost (i.e., with minimal access to the data). The matrix counterpart of this problem (i.e., representation of a large matrix on the basis of a few columns and rows) is referred to as the *pseudoskeleton approximation* [98], where the optimal representation corresponds to the columns and rows that intersect in the submatrix of maximal volume (maximal absolute value of the determinant). Finding the optimal submatrix is computationally hard, but quasioptimal submatrices may be found by heuristic so-called cross-approximation methods that only require a limited, partial exploration of the data matrix. Tucker variants of this approach have been derived in [99]–[101] and are illustrated in Figure 11, while a cross-approximation for the TT format has been derived in [102]. Following a somewhat different idea, a tensor generalization of the CUR decomposition of matrices samples fibers on the basis of statistics derived from the data [103].

MULTIWAY REGRESSION—HIGHER-ORDER PARTIAL LS

MULTIVARIATE REGRESSION

Regression refers to the modeling of one or more dependent variables (responses), Y, by a set of independent data (predictors), X. In the simplest case of conditional mean square estimation (MSE), whereby $\hat{y} = E(y \mid x)$, the response y is a linear combination of the elements of the vector of predictors $\mathbf{x}$; for multivariate data, the multivariate linear regression (MLR) uses a matrix model, $\mathbf{Y} = \mathbf{XP} + \mathbf{E}$, where $\mathbf{P}$ is the matrix of coefficients (loadings) and $\mathbf{E}$ is the residual matrix. The MLR solution gives $\mathbf{P} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ and involves inversion of the moment matrix $\mathbf{X}^T\mathbf{X}$. A common technique to stabilize the inverse of the moment matrix $\mathbf{X}^T\mathbf{X}$ is the principal component regression (PCR), which employs low-rank approximation of $\mathbf{X}$.

MODELING STRUCTURE IN DATA—THE PARTIAL LS

Note that in stabilizing multivariate regression, PCR uses only information in the X variables, with no feedback from the Y variables. The idea behind the partial LS (PLS) method is to account for structure in data by assuming that the underlying system is governed by a small number, R, of specifically constructed latent variables, called scores, that are shared between the X and Y variables; in estimating the number R, PLS compromises between fitting $\mathbf{X}$ and predicting $\mathbf{Y}$. Figure 12 illustrates that the PLS procedure: 1) uses eigenanalysis to perform contraction of the data matrix $\mathbf{X}$ to the principal eigenvector score matrix $\mathbf{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_R]$ of rank R and 2) ensures that the $\boldsymbol{t}_r$ components are maximally correlated with the $\boldsymbol{u}_r$ components in the approximation of the responses $\mathbf{Y}$, this is achieved when the $\mathbf{u}_r$'s are scaled versions of the $\boldsymbol{t}_r$'s. The Y-variables are then regressed on the matrix $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_R]$. Therefore, PLS is a multivariate model with inferential ability that aims to find a representation of $\mathbf{X}$ (or a part of $\mathbf{X}$) that is relevant for predicting $\mathbf{Y}$, using the model

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \sum_{r=1}^{R} \boldsymbol{t}_r \boldsymbol{p}_r^T + \mathbf{E}, \qquad (15)$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F} = \sum_{r=1}^{R} \boldsymbol{u}_r \boldsymbol{q}_r^T + \mathbf{F}. \qquad (16)$$

The score vectors $\boldsymbol{t}_r$ provide an LS fit of $\mathbf{X}$-data, while at the same time, the maximum correlation between $\boldsymbol{t}$ and $\boldsymbol{u}$ scores ensures a good predictive model for Y variables. The predicted responses $\mathbf{Y}_{\text{new}}$ are then obtained from new data $\mathbf{X}_{\text{new}}$ and the loadings $\mathbf{P}$ and $\mathbf{Q}$.

In practice, the score vectors $\boldsymbol{t}_r$, are extracted sequentially, by a series of orthogonal projections followed by the deflation of $\mathbf{X}$. Since the rank of $\mathbf{Y}$ is not necessarily decreased with each new $\boldsymbol{t}_r$, we may continue deflating until the rank of the $\mathbf{X}$-block is exhausted so as to balance between prediction accuracy and model order.

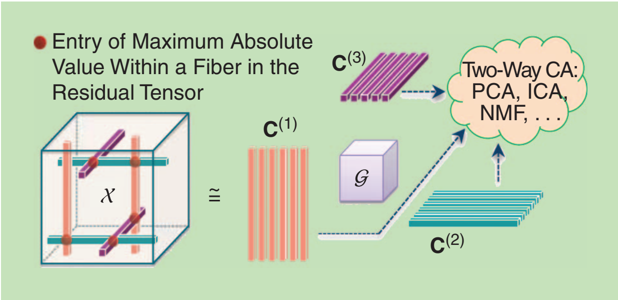
The PLS concept can be generalized to tensors in the following ways:

1) *Unfolding multiway data*. For example, tensors $\mathcal{X}(I \times J \times K)$ and $\mathcal{Y}(I \times M \times N)$ can be flattened into long matrices $\mathbf{X}(I \times JK)$ and $\mathbf{Y}(I \times MN)$ so as to admit matrix-PLS (see Figure 12). However, such flattening prior to standard bilinear PLS obscures the structure in multiway data and compromises the interpretation of latent components.

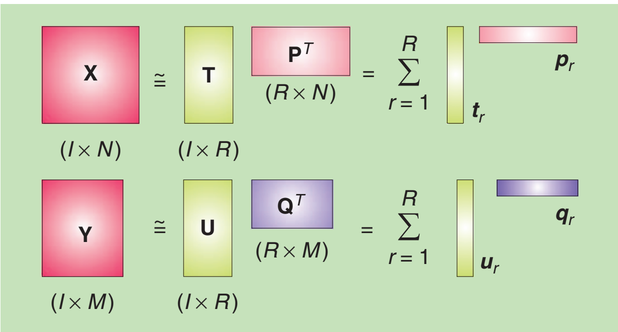
2) *Low-rank tensor approximation*. The so-called N-PLS attempts to find score vectors having maximal covariance with response variables, under the constraints that tensors $\mathcal{X}$ and $\mathcal{Y}$ are decomposed as a sum of rank-1 tensors [104].

3) *A BTD-type approximation*. As in the higher-order PLS (HOPLS) model shown in Figure 13 [105], the use of block terms within HOPLS equips it with additional flexibility, together with a more physically meaningful analysis than unfolding-PLS and N-PLS.

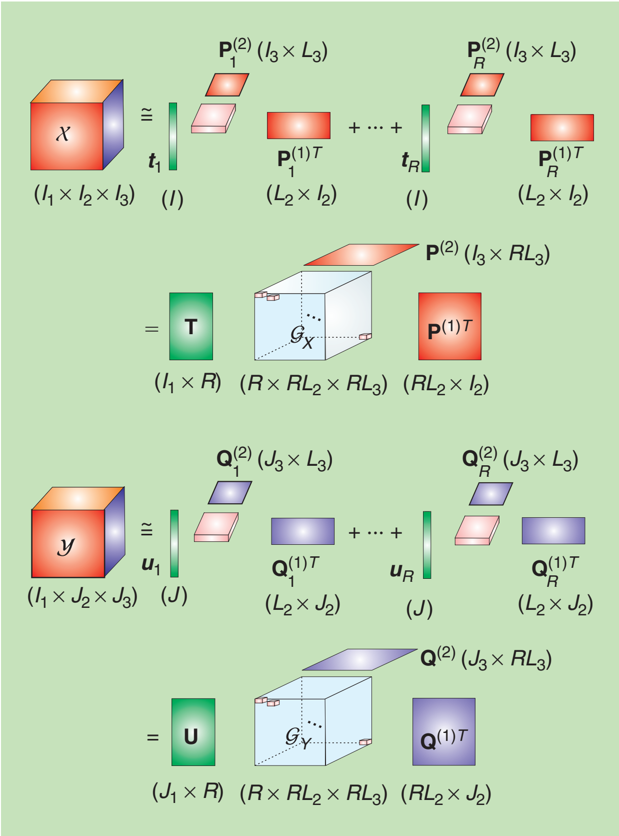
The principle of HOPLS can be formalized as a set of sequential approximate decompositions of the independent tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and the dependent tensor $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$ (with $I_1 = J_1$) so as to ensure maximum similarity (correlation) between the scores $\boldsymbol{t}_r$ and $\boldsymbol{u}_r$ within the matrices $\mathbf{T}$ and $\mathbf{U}$, based on



[FIG11] The Tucker representation through fiber sampling and cross-approximation: the columns of factor matrices are sampled from the fibers of the original data tensor $\mathcal{X}$. Within MWCA, the selected fibers may be further processed using BSS algorithms.



[FIG12] The basic PLS model performs joint sequential low-rank approximation of the matrix of predictors $\mathbf{X}$ and the matrix of responses $\mathbf{Y}$ so as to share (up to the scaling ambiguity) the latent components—columns of the score matrices $\mathbf{T}$ and $\mathbf{U}$. The matrices $\mathbf{P}$ and $\mathbf{Q}$ are the loading matrices for predictors and responses, and $\mathbf{E}$ and $\mathbf{F}$ are the corresponding residual matrices.



[FIG13] The principle of HOPLS for third-order tensors. The core tensors $\mathcal{G}_X$ and $\mathcal{G}_Y$ are block-diagonal. The BTD-type structure allows for the modeling of general components that are highly correlated in the first mode.

$$\mathcal{X} \cong \sum_{r=1}^{R} \mathcal{G}_X^{(r)} \times_1 \boldsymbol{t}_r \times_2 \mathbf{P}_r^{(1)} \cdots \times_N \mathbf{P}_r^{(N-1)} \tag{17}$$

$$\mathcal{Y} \cong \sum_{r=1}^{R} \mathcal{G}_Y^{(r)} \times_1 \boldsymbol{u}_r \times_2 \mathbf{Q}_r^{(1)} \cdots \times_N \mathbf{Q}_r^{(M-1)}. \tag{18}$$

A number of data-analytic problems can be reformulated as either regression or similarity analysis [analysis of variance (ANOVA), autoregressive moving average modeling (ARMA), linear discriminant analysis (LDA), and canonical correlation analysis (CCA)], so that both the matrix and tensor PLS solutions can be generalized across exploratory data analysis.

EXAMPLE 4

The predictive power of tensor-based PLS is illustrated on a real-world example of the prediction of arm movement trajectory from the electrocorticogram (ECoG). Figure 14(a) illustrates the experimental setup, whereby the 3-D arm movement of a monkey was captured by an optical motion capture system with reflective markers affixed to the left shoulder, elbow, wrist, and hand; for full details, see http://neurotycho.org. The predictors (32 ECoG channels) naturally build a fourth-order tensor $\mathcal{X}$ (time × channel_no × epoch_length × frequency) while the movement trajectories for the four markers (response) can be represented as a third-order tensor $\mathcal{Y}$ (time × 3D_marker_position × marker_no). The goal of the training stage is to identify the HOPLS parameters: $\mathcal{G}_X^{(r)}, \mathcal{G}_Y^{(r)}, \mathbf{P}_r^{(n)}, \mathbf{Q}_r^{(n)}$ (see Figure 13). In the test stage, the movement trajectories, $\mathcal{Y}^*$, for the new ECoG data, $\mathcal{X}^*$, are predicted through multilinear projections: 1) the new scores, $\boldsymbol{t}_r^*$, are found from new data, $\mathcal{X}^*$, and the existing model parameters: $\mathcal{G}_X^{(r)}, \mathbf{P}_r^{(1)}, \mathbf{P}_r^{(2)}, \mathbf{P}_r^{(3)}$, and 2) the predicted trajectory is calculated as $\mathcal{Y}^* \approx \sum_{r=1}^{R} \mathcal{G}_Y^{(r)} \times_1 \boldsymbol{t}_r^* \times_2 \mathbf{Q}_r^{(1)} \times_3 \mathbf{Q}_r^{(2)} \times_4 \mathbf{Q}_r^{(3)}$. In the simulations, standard PLS was applied in the same way to the unfolded tensors.

Figure 14(c) shows that although the standard PLS was able to predict the movement corresponding to each marker individually, such a prediction is quite crude as the two-way PLS does not adequately account for mutual information among the four markers. The enhanced predictive performance of the BTD-based HOPLS [the red line in Figure 14(c)] is therefore attributed to its ability to model interactions between complex latent components of both predictors and responses.

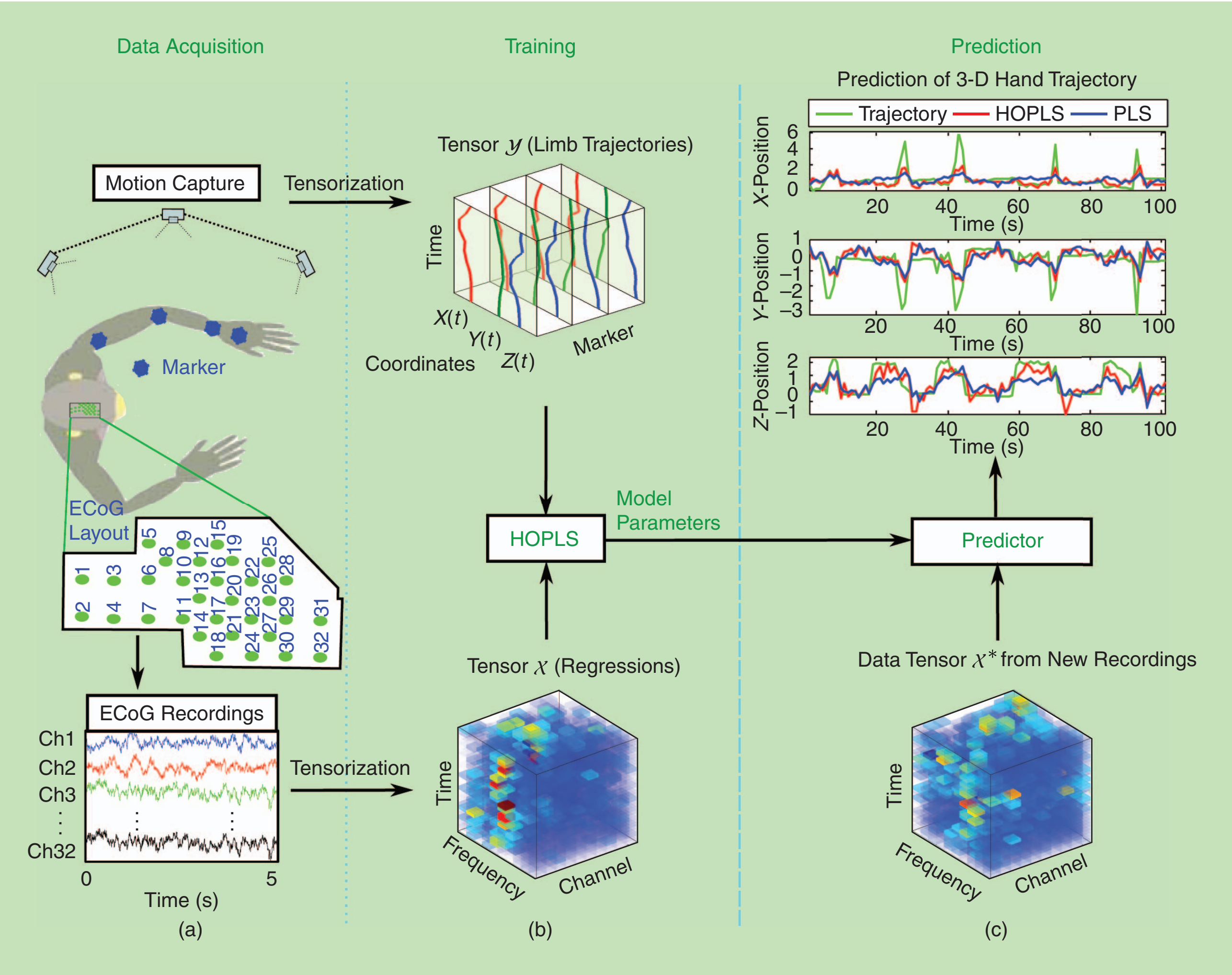
LINKED MULTIWAY COMPONENT ANALYSIS AND TENSOR DATA FUSION

Data fusion concerns the joint analysis of an ensemble of data sets, such as multiple views of a particular phenomenon, where some parts of the scene may be visible in only one or a few data sets. Examples include the fusion of visual and thermal images in low-visibility conditions and the analysis of human electrophysiological signals in response to a certain stimulus but from different subjects and trials; these are naturally analyzed together by means of matrix/tensor factorizations. The coupled nature of the analysis of such multiple data sets ensures that we are able to account for the common factors across the data sets and, at the same time, to guarantee that the individual components are not shared (e.g., processes that are independent of excitations or stimuli/tasks).

The linked multiway component analysis (LMWCA) [106], shown in Figure 15, performs such a decomposition into shared and individual factors and is formulated as a set of approximate joint TKD of a set of data tensors $\mathcal{X}^{(k)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $(k = 1, 2, \ldots, K)$

$$\mathcal{X}^{(k)} \cong \mathcal{G}^{(k)} \times_1 \mathbf{B}^{(1,k)} \times_2 \mathbf{B}^{(2,k)} \cdots \times_N \mathbf{B}^{(N,k)}, \tag{19}$$

where each factor matrix $\mathbf{B}^{(n,k)} = [\mathbf{B}_C^{(n)}, \mathbf{B}_I^{(n,k)}] \in \mathbb{R}^{I_n \times R_n}$ has 1) components $\mathbf{B}_C^{(n)} \in \mathbb{R}^{I_n \times C_n}$ (with $0 \le C_n \le R_n$) that are common (i.e., maximally correlated) to all tensors and 2) components $\mathbf{B}_I^{(n,k)} \in \mathbb{R}^{I_n \times (R_n - C_n)}$ that are tensor specific. The objective is to estimate the common components $\mathbf{B}_C^{(n)}$, the individual components $\mathbf{B}_I^{(n,k)}$, and, via the core tensors $\mathcal{G}^{(k)}$, their mutual interactions. As in MWCA (see the section "Tucker Decomposition"), constraints may be imposed to match data properties [73], [76]. This enables a more general and flexible framework than group ICA and independent vector analysis, which also performs linked analysis of multiple data sets but assume that 1) there exist only common components and 2) the corresponding latent variables are statistically independent [107], [108]. Both are quite stringent and limiting assumptions. As an alternative to TKD, coupled tensor decompositions may be of a polyadic or even block term type [89], [109].



[FIG14] The prediction of arm movement from brain electrical responses. (a) The experiment setup. (b) The construction of the data and response tensors and training. (c) The new data tensor (bottom) and the predicted 3-D arm movement trajectories (*X, Y, Z* coordinates) obtained by tensor-based HOPLS and standard matrix-based PLS (top).

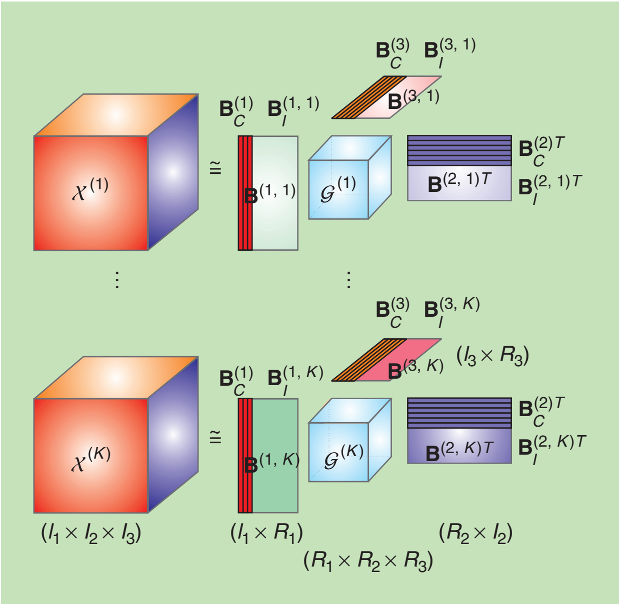
EXAMPLE 5

We employed LWCA for classification based on common and distinct features of natural objects from the ETH-80 database (http://www.d2.mpi-inf.mpg.de/Data sets/ETH80) whereby the discrimination among objects was performed using only the common features. This data set consists of 3,280 images in eight categories, each containing ten objects with 41 views per object. For each category, the training data were organized in two distinct fourth-order $(128 \times 128 \times 3 \times I_4)$ tensors, where $I_4 = 10 \times 41 \times 0.5p$, where p denotes the fraction of training data. LMWCA was applied to these two tensors to find the common and individual features, with the number of common features set to 80% of I_4. In this way, eight sets of common features were obtained for each category. The test sample label was assigned to the category whose common features matched the new sample best (evaluated by canonical correlations) [110]. Figure 16 compares LMWCA with the standard K-nearest neighbors (K-NNs) and LDA classifiers (using 50 principal components as features), all averaged over 50 Monte Carlo runs. The enhanced classification results for LMWCA are attributed to the fact that the classification makes use of only the common components and is not hindered by components that are not shared across objects or views.

SOFTWARE

The currently available software resources for tensor decompositions include:

- The tensor toolbox, a versatile framework for basic operations on sparse and dense tensors, including CPD and Tucker formats [111].
- The TDALAB and TENSORBOX, which provide a user-friendly interface and advanced algorithms for CPD, nonnegative TKD, and MWCA [112], [113].
- The Tensorlab toolbox builds upon the complex optimization framework and offers numerical algorithms for computing the CPD, BTD, and TKD; the toolbox includes a library of constraints (e.g., nonnegativity and orthogonality) and the possibility to combine and jointly factorize dense, sparse, and incomplete tensors [89].



[FIG15] **Coupled TKD for LMWCA. The data tensors have both shared and individual components. Constraints such as orthogonality, statistical independence, sparsity, and nonnegativity may be imposed where appropriate.**

- The N-way toolbox, which includes (constrained) CPD, TKD, and PLS in the context of chemometrics applications [114]; many of these methods can handle constraints (e.g., nonnegativity and orthogonality) and missing elements.
- The TT toolbox, the Hierarchical Tucker toolbox, and the Tensor Calculus library provide tensor tools for scientific computing [115]–[117].
- Code developed for multiway analysis is also available from the Three-Mode Company [118].

CONCLUSIONS AND FUTURE DIRECTIONS

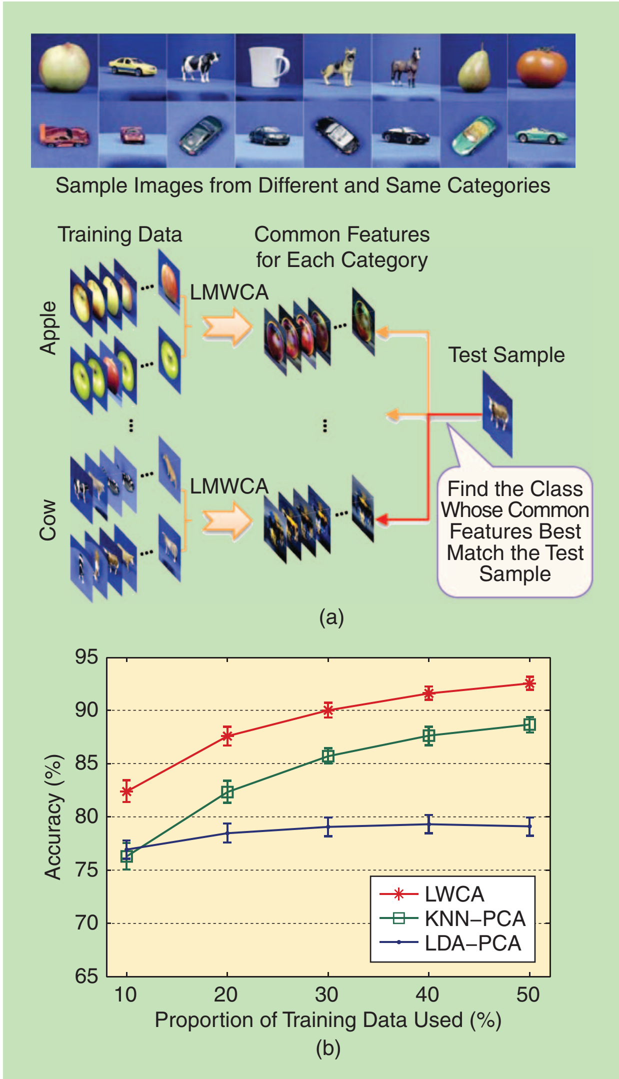
We live in a world overwhelmed by data, from multiple pictures of Big Ben on various social Web links to terabytes of data in multiview medical imaging, while we may also need to repeat the scientific experiments many times to obtain the ground truth. Each snapshot gives us a somewhat incomplete view of the same object and involves different angles, illumination, lighting conditions, facial expressions, and noise.

We have shown that tensor decompositions are a perfect match for exploratory analysis of such multifaceted data sets and have illustrated their applications in multisensor and multimodal signal processing. Our emphasis has been to show that tensor decompositions and multilinear algebra open up completely new possibilities for component analysis, as compared with the flat view of standard two-way methods.

Unlike matrices, tensors are multiway arrays of data samples whose representations are typically overdetermined (fewer parameters in the decomposition than the number of data entries). This gives us an enormous flexibility in finding hidden components in data and the ability to enhance both robustness to noise and tolerance to missing data samples and faulty sensors. We have also discussed multilinear variants of several standard signal processing tools such as multilinear SVD, ICA, NMF, and PLS and have shown that tensor methods can operate in a deterministic way on signals of very short duration.

At present, the uniqueness conditions of standard tensor models are relatively well understood and efficient computation algorithms do exist. However, for future applications, several challenging problems remain to be addressed in more depth.

- A whole new area emerges when several decompositions that operate on different data sets are coupled, as in multiview data where some details of interest are visible in, e.g., only one mode. Such techniques need theoretical support in terms of existence, uniqueness, and numerical properties.
- As the complexity of advanced models increases, their computation requires efficient iterative algorithms, extending beyond the ALS class.



[FIG16] **The classification of color objects belonging to different categories. By using only common features, LMWCA achieves a high classification rate, even when the training set is small. (a) Classification based on LMWCA. (b) Performance comparison.**

- The estimation of the number of components in data and the assessment of their dimensionality would benefit from automation, especially in the presence of noise and outliers.
- Both new theory and algorithms are needed to further extend the flexibility of tensor models, e.g., for the constraints to be combined in many ways and tailored to the particular signal properties in different modes.
- Work on efficient techniques for saving and/or fast processing of ultra-large-scale tensors is urgent; these now routinely occupy terabytes, and will soon require petabytes of memory.
- Tools for rigorous performance analysis and rule of thumb performance bounds need to be further developed across tensor decomposition models.
- Our discussion has been limited to tensor models in which all entries take values independently of one another. Probabilistic versions of tensor decompositions incorporate prior knowledge about complex variable interaction, various data alphabets, or noise distributions, and so promise to model data more accurately and efficiently [119], [120].
- The future computational, visualization, and interpretation tools will be important next steps in supporting the different communities working on large-scale and big data analysis problems.

It is fitting to conclude with a quote from the French novelist Marcel Proust: "The voyage of discovery is not in seeking new landscapes but in having new eyes." We hope to have helped to bring to the eyes of the signal processing community the multidisciplinary developments in tensor decompositions and to have shared our enthusiasm about tensors as powerful tools to discover new landscapes.

AUTHORS

Andrzej Cichocki (cia@brain.riken.jp) received the Ph.D. and Dr.Sc. (habilitation) degrees all in electrical engineering from the Warsaw University of Technology, Poland. He is currently a senior team leader of the Laboratory for Advanced Brain Signal Processing at RIKEN Brain Science Institute, Japan, and a professor at the Systems Research Institute, Polish Academy of Science, Poland. He has authored more than 400 publications and four monographs in the areas of signal processing and computational neuroscience. He is an associate editor of *IEEE Transactions on Signal Processing* and *Journal of Neuroscience Methods*.

Danilo P. Mandic (d.mandic@imperial.ac.uk) is a professor of signal processing at Imperial College London, United Kingdom, and has been working in the area of nonlinear and multidimensional adaptive signal processing and time-frequency analysis. His publication record includes two research monographs, *Recurrent Neural Networks for Prediction* and *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*, an edited book, *Signal Processing for Information Fusion*, and more than 200 publications on signal and image processing. He has been a guest professor at KU Leuven, Belgium, and a frontier researcher at RIKEN Brain Science Institute, Tokyo, Japan.

Anh Huy Phan (phan@brain.riken.jp) received the Ph.D. degree from the Kita Kyushu Institute of Technology, Japan, in 2011. He worked as a deputy head of the Research and Development Department, Broadcast Research and Application Center, Vietnam Television, and is currently a research scientist at the Laboratory for Advanced Brain Signal Processing and a visiting research scientist at the Toyota Collaboration Center, RIKEN Brain Science Institute, Japan. He has served on the editorial board of *International Journal of Computational Mathematics*. His research interests include multilinear algebra, tensor computation, blind source separation, and brain–computer interfaces.

Cesar F. Caiafa (ccaiafa@gmail.com) received the Ph.D. degree in engineering from the Faculty of Engineering, University of Buenos Aires, in 2007. He is currently an adjunct researcher with the Argentinean Radioastronomy Institute (IAR)—CONICET and an assistant professor with Faculty of Engineering, the University of Buenos Aires. He is also a visiting scientist at the Laboratory for Advanced Brain Signal Processing, BSI—RIKEN, Japan.

Guoxu Zhou (zhouguoxu@brain.riken.jp) received the Ph.D. degree in intelligent signal and information processing from the South China University of Technology, Guangzhou, in 2010. He is currently a research scientist at the Laboratory for Advanced Brain Signal Processing at RIKEN Brain Science Institute, Japan. His research interests include statistical signal processing, tensor analysis, intelligent information processing, and machine learning.

Qibin Zhao (qbzhao@brain.riken.jp) received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2009. He is currently a research scientist at the Laboratory for Advanced Brain Signal Processing in RIKEN Brain Science Institute, Japan, and a visiting research scientist in the BSI Toyota Collaboration Center, RIKEN-BSI. His research interests include multiway data analysis, brain–computer interface, and machine learning.

Lieven De Lathauwer (Lieven.DeLathauwer@kuleuven-kulak.be) received the Ph.D. degree from the Faculty of Engineering, KU Leuven, Belgium, in 1997. From 2000 to 2007, he was a research associate with the Centre National de la Recherche Scientifique, France. He is currently a professor with KU Leuven. He is affiliated with the group Science, Engineering, and Technology of Kulak, the Stadius Center for Dynamical Systems, Signal Processing, and Data Analytics of the Electrical Engineering Department (ESAT), and iMinds Future Health Department. He is an associate editor of *SIAM Journal on Matrix Analysis and Applications* and was an associate editor of *IEEE Transactions on Signal Processing*. His research focuses on the development of tensor tools for engineering applications.

REFERENCES

[1] F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *J. Math. Phys.*, vol. 7, no. 1, pp. 39–79, 1927.

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

[3] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices," in *Contributions to Mathematical Psychology*, H. Gulliksen and N. Frederiksen, Eds. New York: Holt, Rinehart and Winston, 1964, pp. 110–127.

[4] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sept. 1966.

[5] J. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, Sept. 1970.

[6] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Pap. Phonet.*, vol. 16, pp. 1–84, 1970.

[7] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*. Hoboken, NJ: Wiley, 2004.

[8] P. Kroonenberg, *Applied Multiway Data Analysis*. Hoboken, NJ: Wiley, 2008.

[9] C. Nikias and A. Petropulu, *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[10] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," in *IEE Proc. F (Radar and Signal Processing)*, vol. 140, no. 6, IET, 1993, pp. 362–370.

[11] P. Comon, "Independent component analysis: A new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[12] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. New York, Academic, 2010.

[13] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. Hoboken, NJ: Wiley, 2003.

[14] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[15] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[16] G. Beylkin and M. Mohlenkamp, "Algorithms for numerical analysis in high dimensions," *SIAM J. Sci. Comput.*, vol. 26, no. 6, pp. 2133–2159, 2005.

[17] J. Ballani, L. Grasedyck, and M. Kluge, "Black box approximation of tensors in hierarchical Tucker format," *Linear Algebr. Appl.*, vol. 433, no. 2, pp. 639–657, 2011.

[18] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[19] N. Sidiropoulos, R. Bro, and G. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Trans. Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[20] N. Sidiropoulos, G. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Trans. Signal Processing*, vol. 48, no. 3, pp. 810–823, 2000.

[21] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Hoboken, NJ: Wiley, 2009.

[22] J. Landsberg, *Tensors: Geometry and Applications*. AMS, 2012.

[23] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus* (ser. Springer series in computational mathematics). Heidelberg: Springer, 2012, vol. 42.

[24] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE Trans. Knowledge Data Eng.*, vol. 21, no. 1, pp. 6–20, 2009.

[25] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sept. 2009.

[26] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemomet.*, vol. 23, no. 7–8, pp. 393–405, 2009.

[27] H. Lu, K. Plataniotis, and A. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognit.*, vol. 44, no. 7, pp. 1540–1551, 2011.

[28] M. Mørup, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *Wiley Interdisc. Rew.: Data Mining Knowled. Discov.*, vol. 1, no. 1, pp. 24–40, 2011.

[29] B. Khoromskij, "Tensors-structured numerical methods in scientific computing: Survey on recent advances," *Chemomet. Intell. Lab. Syst.*, vol. 110, no. 1, pp. 1–19, 2011.

[30] L. Grasedyck, D. Kressner, and C. Tobler, "A literature survey of low-rank tensor approximation techniques," *CGAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013.

[31] P. Comon, "Tensors: A brief introduction," *IEEE Signal Processing Mag.*, vol. 31, no. 3, pp. 44–53, May 2014.

[32] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[33] J. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebr. Appl.*, vol. 18, no. 2, pp. 95–138, 1977.

[34] I. Domanov and L. De Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part I: Basic results and uniqueness of one factor matrix and part II: Uniqueness of the overall decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 855–903, 2013.

[35] M. Elad, P. Milanfar, and G. H. Golub, "Shape from moments—An estimation theory perspective," *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1814–1829, 2004.

[36] N. Sidiropoulos, "Generalizing Caratheodory's uniqueness of harmonic parameterization to N dimensions," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1687–1690, 2001.

[37] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and É. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[38] F. Miwakeichi, E. Martnez-Montes, P. Valds-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, "Decomposing EEG data into space–time–frequency components using parallel factor analysis," *NeuroImage*, vol. 22, no. 3, pp. 1035–1045, 2004.

[39] M. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. European Conf. on Computer Vision (ECCV)*, Copenhagen, Denmark, May 2002, vol. 2350, pp. 447–460.

[40] M. Hirsch, D. Lanman, G.Wetzstein, and R. Raskar, "Tensor displays," in *Proc. Int. Conf. Computer Graphics and Interactive Techniques, SIGGRAPH 2012*, Los Angeles, CA, USA, Aug. 5-9, 2012, *Emerging Technologies Proc.*, 2012, pp. 24–42.

[41] J. Håstad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[42] M. Timmerman and H. Kiers, "Three mode principal components analysis: Choosing the numbers of components and sensitivity to local optima," *Br. J. Math. Stat. Psychol.*, vol. 53, no. 1, pp. 1–16, 2000.

[43] E. Ceulemans and H. Kiers, "Selecting among three-mode principal component models of different types and complexities: A numerical convex-hull based method," *Br. J. Math Stat Psychol.*, vol. 59, no. 1, pp. 133–150, May 2006.

[44] M. Mørup and L. K. Hansen, "Automatic relevance determination for multiway models," *J. Chemomet., Special Issue: In Honor of Professor Richard A. Harshman*, vol. 23, no. 7–8, pp. 352–363, 2009.

[45] N. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays," *J. Chemomet.*, vol. 14, no. 3, pp. 229–239, 2000.

[46] T. Jiang and N. D. Sidiropoulos, "Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models," *IEEE Trans. Signal Processing*, vol. 52, no. 9, pp. 2625–2636, 2004.

[47] L. De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 3, pp. 642–666, 2006.

[48] A. Stegeman, "On uniqueness conditions for CANDECOMP/PARAFAC and INDSCAL with full column rank in one mode," *Linear Algebr. Appl.*, vol. 431, no. 1–2, pp. 211–227, 2009.

[49] E. Sanchez and B. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemomet.*, vol. 4, no. 1, pp. 29–45, 1990.

[50] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition," *SIAM Anal. Appl.*, vol. 35, no. 2, pp. 636–660, 2014.

[51] S. Vorobyov, Y. Rong, N. Sidiropoulos, and A. Gershman, "Robust iterative fitting of multilinear models," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2678–2689, 2005.

[52] X. Liu and N. Sidiropoulos, "Cramér-Rao lower bounds for low-rank decomposition of multidimensional arrays," *IEEE Trans. Signal Processing*, vol. 49, no. 9, pp. 2074–2086, Sept. 2001.

[53] P. Tichavský, A.-H. Phan, and Z. Koldovský, "Cramér-Rao-induced bounds for CANDECOMP/PARAFAC tensor decomposition," *IEEE Trans. Signal Processing*, vol. 61, no. 8, pp. 1986–1997, 2013.

[54] B. Chen, S. He, Z. Li, and S. Zhang, "Maximum block improvement and polynomial optimization," *SIAM J. Optim.*, vol. 22, no. 1, pp. 87–107, 2012.

[55] A. Uschmajew, "Local convergence of the alternating least squares algorithm for canonical tensor approximation," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 2, pp. 639–652, 2012.

[56] M. J. Mohlenkamp, "Musings on multilinear fitting," *Linear Algebr. Appl.*, vol. 438, no. 2, pp. 834–852, 2013.

[57] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[58] P. Paatero, "The multilinear engine: A table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model," *J. Computat. Graph. Stat.*, vol. 8, no. 4, pp. 854–888, Dec. 1999.

[59] E. Acar, D. Dunlavy, T. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemomet. Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011.

[60] A.-H. Phan, P. Tichavský, and A. Cichocki, "Low complexity damped Gauss-Newton algorithms for CANDECOMP/PARAFAC," *SIAM J. Matrix Anal. Appl. (SIMAX)*, vol. 34, no. 1, pp. 126–147, 2013.

[61] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical Polyadic Decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, 2013.

[62] V. de Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 1084–1127, Sept. 2008.

[63] W. Krijnen, T. Dijkstra, and A. Stegeman, "On the non-existence of optimal solutions and the occurrence of "degeneracy" in the CANDECOMP/PARAFAC model," *Psychometrika*, vol. 73, no. 3, pp. 431–439, 2008.

[64] M. Sørensen, L. De Lathauwer, P. Comon, S. Icart, and L. Deneire, "Canonical Polyadic Decomposition with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 4, pp. 1190–1213, 2012.

[65] M. Sørensen and L. De Lathauwer, "Blind signal separation via tensor decomposition with Vandermonde factor: Canonical polyadic decomposition," *IEEE Trans. Signal Processing*, vol. 61, no. 22, pp. 5507–5519, Nov. 2013.

[66] G. Zhou and A. Cichocki, "Canonical Polyadic Decomposition based on a single mode blind source separation," *IEEE Signal Processing Lett.*, vol. 19, no. 8, pp. 523–526, 2012.

[67] L.-H. Lim and P. Comon, "Nonnegative approximations of nonnegative tensors," *J. Chemomet.*, vol. 23, nos. 7–8, pp. 432–441, 2009.

[68] A. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, no. 5, pp. 1136–1155, 1996.

[69] R. Roy and T. Kailath, "ESPRIT—estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, no. 7, pp. 984–995, 1989.

[70] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-$R_1, R_2, \ldots, R_N$ approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

[71] B. Savas and L.-H. Lim, "Quasi-Newton methods on Grassmannians and multilinear approximations of tensors," *SIAM J. Sci. Comput.*, vol. 32, no. 6, pp. 3352–3393, 2010.

[72] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer, "Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme," *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 1, pp. 115–135, 2011.

[73] G. Zhou and A. Cichocki, "Fast and unique Tucker decompositions via multiway blind source separation," *Bull. Polish Acad. Sci.*, vol. 60, no. 3, pp. 389–407, 2012.

[74] A. Cichocki, "Generalized component analysis and blind source separation methods for analyzing mulitchannel brain signals," in *Statistical and Process Models for Cognitive Neuroscience and Aging*. Lawrence Erlbaum Associates, 2007, pp. 201–272.

[75] M. Haardt, F. Roemer, and G. D. Galdo, "Higher-order SVD based subspace estimation to improve the parameter estimation accuracy in multi-dimensional harmonic retrieval problems," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3198–3213, July 2008.

[76] A.-H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional data sets," *Nonlinear Theory Appl., IEICE*, vol. 1, no. 1, pp. 37–68, 2010.

[77] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part I and II," *SIAM J. Matrix Anal. Appl. (SIMAX) Special Issue on Tensor Decompositions and Applications*, vol. 30, no. 3, pp. 1022–1066, 2008.

[78] L. De Lathauwer, "Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(L_r, L_r, 1)$ terms," *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 4, pp. 1451–1474, 2011.

[79] L. De Lathauwer, "Block component analysis: A new concept for blind source separation," in *Proc. 10th Int. Conf. LVA/ICA, Tel Aviv,* Israel, Mar. 12–15, 2012, pp. 1–8.

[80] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[81] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[82] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[83] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications,* vol. 20. New York: Cambridge Univ. Press, 2012, p. 12.

[84] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Trans. Image Processing*, vol. 21, no. 2, pp. 494–504, 2012.

[85] C. Caiafa and A. Cichocki, "Computing sparse representations of multidimensional signals using Kronecker bases," *Neural Computat.*, vol. 25, no. 1, pp. 186–220, 2013.

[86] C. Caiafa and A. Cichocki, "Multidimensional compressed sensing and their applications," *WIREs Data Mining Knowled. Discov.*, vol. 3, no. 6, pp. 355–380, 2013.

[87] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Prob.*, vol. 27, no. 2, pp. 1–19, 2011.

[88] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. K. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, Mar. 2014.

[89] L. Sorber, M. Van Barel, and L. De Lathauwer. (2014, Jan.). Tensorlab v2.0. [Online]. Available: www.tensorlab.net

[90] N. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Processing Lett.*, vol. 19, no. 11, pp. 757–760, 2012.

[91] D. Foster, K. Amano, S. Nascimento, and M. Foster, "Frequency of metamerism in natural scenes," *J. Opt. Soc. Amer. A*, vol. 23, no. 10, pp. 2359–2372, 2006.

[92] A. Cichocki, "Era of big data processing: A new approach via tensor networks and tensor decompositions (invited talk)," in *Proc. 2013 Int. Workshop on Smart Info-Media Systems in Asia, SISA-2013*, Nagoya, Japan, Oct. 1, 2013, 2013, 30 pages. [Online]. Available: http://arxiv.org/pdf/1403.2048.pdf

[93] R. Orus, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *J. Chem. Phys.*, 2013.

[94] J. Salmi, A. Richter, and V. Koivunen, "Sequential unfolding SVD for tensors with applications in array signal processing," *IEEE Trans. Signal Processing*, vol. 57, no. 12, pp. 4719–4733, 2009.

[95] A.-H. Phan and A. Cichocki, "PARAFAC algorithms for large-scale problems," *Neurocomputing*, vol. 74, no. 11, pp. 1970–1984, 2011.

[96] S. K. Suter, M. Makhynia, and R. Pajarola, "TAMRESH: Tensor approximation multiresolution hierarchy for interactive volume visualization," *Comput. Graph. Forum*, vol. 32, no. 3, pp. 151–160, 2013.

[97] D. Nion and N. Sidiropoulos, "Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor," *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2299–2310, June 2009.

[98] S. A. Goreinov, N. L. Zamarashkin, and E. E. Tyrtyshnikov, "Pseudo-skeleton approximations by matrices of maximum volume," *Math. Notes*, vol. 62, no. 4, pp. 515–519, 1997.

[99] C. Caiafa and A. Cichocki, "Generalizing the column-row matrix decomposition to multi-way arrays," *Linear Algebr. Appl.*, vol. 433, no. 3, pp. 557–573, 2010.

[100] S. A. Goreinov, "On cross approximation of multi-index array," *Doklady Math.*, vol. 420, no. 4, pp. 404–406, 2008.

[101] I. Oseledets, D. V. Savostyanov, and E. Tyrtyshnikov, "Tucker dimensionality reduction of three-dimensional arrays in linear time," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 939–956, 2008.

[102] I. Oseledets and E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebr. Appl.*, vol. 432, no. 1, pp. 70–88, 2010.

[103] M. W. Mahoney, M. Maggioni, and P. Drineas, "Tensor-CUR decompositions for tensor-based data," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 957–987, 2008.

[104] R. Bro, "Multiway calibration. Multilinear PLS," *J. Chemomet.*, vol. 10, no. 1, pp. 47–61, 1996.

[105] Q. Zhao, C. Caiafa, D. Mandic, Z. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, "Higher-order partial least squares (HOPLS): A generalized multilinear regression method," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 35, no. 7, pp. 1660–1673, 2013.

[106] A. Cichocki, "Tensors decompositions: New concepts for brain data analysis?" *J. Control, Measure*., Syst. Integr. (SICE), vol. 47, no. 7, pp. 507–517, 2011.

[107] V. Calhoun, J. Liu, and T. Adali, "A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data," *Neuroimage*, vol. 45, pp. 163–172, 2009.

[108] Y.-O. Li, T. Adali, W. Wang, and V. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Processing*, vol. 57, no. 10, pp. 3918–3929, Oct. 2009.

[109] E. Acar, T. Kolda, and D. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," in *Proc. Mining and Learning with Graphs, (MLG'11)*, San Diego, CA, August 2011.

[110] G. Zhou, A. Cichocki, S. Xie, and D. Mandic. (2013). Beyond canonical correlation analysis: Common and individual features analysis. *IEEE Trans. Pattern Anal. Mach. Intell*. [Online]. Available: http://arxiv.org/abs/1212.3913

[111] B. Bader, T. G. Kolda et al. (2012, Feb.). MATLAB tensor toolbox version 2.5. [Online]. Available: http://www.sandia.gov/ tgkolda/TensorToolbox/

[112] G. Zhou and A. Cichocki. (2013). TDALAB: Tensor decomposition laboratory, LABSP, Wako-shi, Japan. [Online]. Available: http://bsp.brain.riken.jp/TDALAB/

[113] A.-H. Phan, P. Tichavský, and A. Cichocki. (2012). TENSORBOX: A MATLAB package for tensor decomposition, LABSP, RIKEN, Japan. [Online]. Available: http://www.bsp.brain.riken.jp/ phan/tensorbox.php

[114] C. Andersson and R. Bro. (2000). The N-way toolbox for MATLAB. [Online]. *Chemomet. Intell. Lab. Syst*., 52(1), pp. 1–4, 2000. Available: http://www.models.life.ku.dk/nwaytoolbox

[115] I. Oseledets. (2012). TT-toolbox 2.2. [Online]. Available: https://github.com/oseledets/TT-Toolbox

[116] D. Kressner and C. Tobler. (2012). htucker—A MATLAB toolbox for tensors in hierarchical Tucker format. MATHICSE, EPF Lausanne. [Online]. Available: http://anchp.epfl.ch/htucker

[117] M. Espig, M. Schuster, A. Killaitis, N. Waldren, P. Wähnert, S. Handschuh, and H. Auer. (2012). Tensor calculus library. [Online]. Available: http://gitorious.org/tensorcalculus

[118] P. Kroonenberg. The three-mode company: A company devoted to creating three-mode software and promoting three-mode data analysis. [Online]. Available: http://three-mode.leidenuniv.nl/.

[119] Z. Xu, F. Yan, and A. Qi, "Infinite Tucker decomposition: Nonparametric Bayesian models for multiway data analysis," in *Proc. 29th Int. Conf. Machine Learning (ICML-12), ser. ICML'12*. Omnipress, July 2012, pp. 1023–1030.

[120] K. Yilmaz and A. T. Cemgil, "Probabilistic latent tensor factorisation," in *Proc. Int. Conf. Latent Variable Analysis and Signal Separation, cPCI-S*, 2010, vol. 6365, pp. 346–353.