The Concept of Tensorization

Otto Debals, *Student Member, IEEE* and Lieven De Lathauwer, *Fellow, IEEE*

***Abstract*—Multiway arrays, also known as tensors, are ubiquitous in domains such as data mining and signal processing. The powerful dimension reduction properties and uniqueness properties of tensor tools, to name a few, have seen a wide applicability with respect to generic problems such as component retrieval and compression. Still, a tensor is required to enable the use of tensor tools. As often only vector or matrix data is available, one can first map the vector or matrix data to a tensor. Various techniques have made use of such a *tensorization* step, whether or not knowingly. Rather than focusing on tensor tools, we will zoom in on this concept of tensorization. We present an overview of various tensorization techniques and show how tensorization is fundamental in applications such as independent component analysis, neural network training, time–frequency analysis and graph clustering. Furthermore, we illustrate how conceptual problems from engineering and mathematics can be well expressed in a tensor-based manner, such as the approximation of non-linear functions with polynomials, matrix–matrix multiplication and systems of polynomial equations.**

***Index Terms*—Tensorization, matrix factorization, tensor decomposition, canonical polyadic decomposition, multiway data, Hankel, Löwner, time–frequency representations, time–scale representations, wavelets, monomial relations, higher-order statistics, moments, cumulants, adjacency matrix, score functions, polarization, blind source separation, independent component analysis, harmonic retrieval, topic modeling, classification, clustering, neural networks, graphs**

I. Introduction

Tensor tools have been applied widely across various domains such as scientific computing, signal and image processing, machine learning and statistics. Tensors are generalizations of vectors and matrices and provide natural structures to store and represent multiway data. Multilinear algebra forms the mathematical backbone of tensors and tensor tools, encompassing far more rich concepts compared to its linear counterpart. This has allowed multilinear tensor models to surpass the flat-view bilinear matrix models and has enabled novel applications dealing with processes such as component retrieval, compression, classification and prediction. While researchers such as Hitchcock [1], [2] and Cattell [3], [4] have provided the foundations of multilinear algebra in the early 20th century, the spark in interest only came from the 60's on thanks to results attributed to Tucker [5], Harshman [6], [7], Carroll and Chang [8] and Kruskal [9] which mainly originated from the domains of psychometrics and chemometrics.

Various influential survey papers and books have appeared discussing models, algorithms and applications of tensor tools, e.g., [10]–[14]. A large class of tensor tools consists of tensor decompositions such as the canonical polyadic decomposition (CPD), the tensor train decomposition (TT) and the multilinear singular value decomposition (MLSVD) or Tucker decomposition. These tensor tools obviously require a tensor. Often, a tensor is readily available, whether in a natural way (e.g., a physical quantity measured along the three common spatial coordinates) or by experiment design (e.g., by repeating an experiment several times and by stacking the obtained data matrices). In some cases, however, only a single vector or matrix is available. Still, an increasing number of techniques from various scientific domains are appearing in which tensor tools are applied given only vector or matrix data. These techniques have been used to separate independent signals [15], [16] or constant modulus signals [17] in signal processing, to identify graph clusters in graph analysis [18], to assign topics to documents in topic modeling [19], [20] and to train neural networks in machine learning [21], [22], to name only a few examples. In each of these applications, only a single vector or matrix is available. Instead of solely being adopted by multiway data, the power and flexibility of tensor tools have seen their way to one-way and two-way data as well.

One might wonder how this apparent paradox can be explained and how tensor tools can be applied given only vector or matrix data. Fundamental in these techniques is the paradigm of mapping the given vector or matrix data to a tensor first, yielding a *tensorization* step in which the lower-order data is tensorized to a tensor. This tensorization procedure is often not well described and sometimes not even recognized. Tensorization effectively translates the assumed vector or matrix model to a multiway model. The obtained tensor typically admits properties such as low-rankness or diagonality for specific sets of vectors or matrices. Various tensor tools build on such tensor properties. For example, given a set of mutually statistically independent signals, the corresponding fourth-order cumulant is diagonal which allows the use of the CPD in independent component analysis (ICA) [16]. Hence, an investigation of the applied tensorization technique is crucial to be able to validate the use of tensor tools in the subsequent step.

In this overview paper, we provide a tensorization framework and we survey different tensorization techniques. The goal consists of presenting an insightful view on the concept

This research is funded by (1) a Ph.D. grant of the Agency for Innovation by Science and Technology (IWT), (2) Research Council KU Leuven: C1 project c16/15/059-nD, (3) F.W.O.: projects G.0830.14N and G.0881.14N, (4) the Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO II, Dynamical systems, control and optimization, 2012-2017), (5) EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (no. 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

O. Debals and L. De Lathauwer are with the Group of Science, Engineering and Technology, KU Leuven Kulak, E. Sabbelaan 53, B-8500 Kortrijk, Belgium and with the Department of Electrical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. (e-mail: otto.debals@kuleuven.be, lieven.delathauwer@kuleuven.be).

Manuscript received XXXX YY, 2016; revised XXXX YY, 2016.

of tensorization, illustrating the power of tensor tools given only vector or matrix data. Results regarding tensorization have appeared rather disparately in the literature. We will connect different tensorization techniques and show that theoretical results and algorithms can be interchanged between these techniques and, thereby, between scientific domains. For example, as we will see, time–frequency and time–scale representations [23]–[25] from signal analysis on the one hand, and segmentation from array processing and function approximation [26], [27] on the other hand, have common grounds such that results from the latter can be passed to the former. Throughout the paper, we will show how the different techniques can be used in various applications such as deterministic blind signal separation, topic modeling, graph clustering, independent component analysis and neural network training.

Together with tensor construction often comes the recognition of a tensor. With tensor recognition, we mean that obtaining a tensor sometimes boils down to recognizing that a problem already involves a tensor. The tensor may be there only implicitly, and seeing it is not always obvious. We will see that multilinear functions and polynomials can be uniquely represented by a tensor. It follows that a number of fundamental mathematical and engineering concepts can very well be formulated using tensor-based expressions. For example, while the concept of linearization is related to linear algebra, the polynomial-based approximation of a non-linear function corresponds to multilinear algebra. Furthermore, topics such as matrix–matrix multiplication, global polynomial optimization and systems of polynomial equations all have clear links with tensors, as will be discussed. The link between multilinear functions and tensors even allows us to develop a framework based on a process called polarization from which new tensorization techniques can originate.

The paper is organized as follows. In Section II, a number of preliminaries are provided. Section III then discusses two common ways how measurements or evaluations readily yield multiway data which can be represented by a tensor. It is clear that multiway data can often be seen as evaluations of multivariate functions; the latter are discussed in Section IV. Together with function evaluation, also function differentiation is covered. Instead of considering tensors modeling multiway data, Section V considers tensors representing objects or operators such as multilinear functions and polynomials. The two latter sections form the theoretical spine of tensorization techniques, which are reviewed in Sections VI and VII. A distinction is made between those tensorization techniques that are able to map a single vector to a tensor (Section VI), and those techniques that map a given matrix to a tensor (Section VII). Section VIII then provides further discussions and pointers to future work. Figure 1 illustrates the organization of the paper.

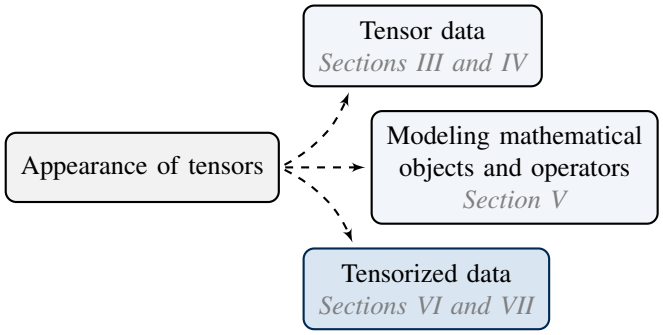


Fig. 1. Organization of the paper regarding the appearance of tensors.

II. Preliminaries on (multi)linear algebra and multiway analysis

Basic notations are given in Section II-A while preliminaries on functions are discussed in Section II-B. Matrix factorizations and tensor decompositions are then covered in Sections II-C and II-D, respectively.

A. Basic notation and matrix/tensor operations

We will denote scalars, vectors, matrices and tensors by lower case (e.g., a), bold lower case (e.g., $\mathbf{a}$), bold upper case (e.g., $\mathbf{A}$) and calligraphic (e.g., $\mathcal{A}$) letters, respectively. Functions are denoted by underlined letters, e.g., a scalar function $\underline{f}$ and vector function $\underline{\mathbf{f}}$. An Nth-order tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multiway array with numerical values $a_{i_1 i_2 \cdots i_N} = \mathcal{A}(i_1, i_2, \ldots, i_N)$, with $\mathbb{K}$ denoting either $\mathbb{R}$ or $\mathbb{C}$. The mode-n vectors of $\mathcal{A}$ are constructed by fixing all but one index. The ith column of a matrix $\mathbf{A}$ is denoted with $\mathbf{a}_i$. Sets are indexed by superscripts[1] within parentheses, e.g., $\left\{\mathbf{A}^{(n)}\right\}_{n=1}^{N}$. The mode-$n$ unfolding of $\mathcal{A}$ is a matrix $\mathbf{A}_{(n)}$ with the mode-n vectors as its columns. The vectorization operator $\text{vec}\,(\mathcal{A})$ stacks all mode-1 vectors into a column vector. On the other hand, a vector $\mathbf{a} \in \mathbb{K}^N$ or matrix $\mathbf{A} \in \mathbb{K}^{M \times N}$ can also be folded to a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times \ldots I_D}$ with $N = \prod_{d=1}^{D} I_d$ and $MN = \prod_{d=1}^{D} I_d$, respectively.

The complex conjugate, transpose, conjugated transpose, inverse, transposed inverse and pseudo-inverse are denoted by $\bar{\cdot}$, $\cdot^{\text{T}}$, $\cdot^{\text{H}}$, $\cdot^{-1}$, $\cdot^{-\text{T}}$ and $\cdot^{\dagger}$, respectively. $r(\mathbf{A})$ returns the rank of $\mathbf{A}$ while $\det(\mathbf{A})$ returns the determinant of a square matrix $\mathbf{A}$. The column-wise and row-wise concatenation of vectors and/or matrices $\mathbf{A}$ and $\mathbf{B}$ is defined by $\mathbf{X} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}$ and $\mathbf{X} = \begin{bmatrix}\mathbf{A}; & \mathbf{B}\end{bmatrix}$, respectively, with the latter being equal to $\mathbf{X} = \begin{bmatrix}\mathbf{A}^{\text{T}} & \mathbf{B}^{\text{T}}\end{bmatrix}^{\text{T}}$. The vectors $\mathbf{1}_N$ and $\mathbf{0}_N$ denote the $N \times 1$ vectors of all ones and zeros, respectively. A similar definition holds for the matrices $\mathbf{1}_{M \times N}$ and $\mathbf{0}_{M \times N}$ of size $M \times N$. $\mathbf{I}_N$ is defined as the identity matrix of size $N \times N$. The canonical basis vector $\mathbf{e}_i$ contains a one in the ith entry and zero elsewhere.

The inner product between two tensors $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\langle \mathcal{A}, \mathcal{B} \rangle = \text{vec}\,(\mathcal{B})^{\text{H}} \text{vec}\,(\mathcal{A})$. The tensor product or outer product of two tensors $\mathcal{A} \in \mathbb{K}^{I_1 \times \cdots \times I_M}$ and $\mathcal{B} \in \mathbb{K}^{J_1 \times \cdots \times J_N}$ is the tensor $\mathcal{C} = \mathcal{A} \otimes \mathcal{B} \in \mathbb{K}^{I_1 \times \cdots \times I_M \times J_1 \times \cdots \times J_N}$ with entries $c_{i_1 \ldots i_M j_1 \ldots j_N} = a_{i_1 \ldots i_M} b_{j_1 \ldots j_N}$ for all indices. The Hadamard or element-wise product and the Kronecker, column-wise

[1] Superscripts typically indicate a set of objects of different types (such as factor matrices with different dimensions), while subscripts typically indicate a set of objects of similar types which can be naturally stacked (such as vectors of equal lengths).

Khatri–Rao and row-wise Khatri–Rao products are denoted with $*$, $\otimes$, $\odot$ and $\odot^{\mathrm{T}}$, respectively. The columns of $\mathbf{A} \odot \mathbf{B}$ (resp., rows of $\mathbf{A} \odot^{\mathrm{T}} \mathbf{B}$) are the pairwise Kronecker products of the columns (resp., rows) of $\mathbf{A}$ and $\mathbf{B}$. Note that the outer and Kronecker product are related through a vectorization: $\mathrm{vec}\,(\mathbf{a} \otimes \mathbf{b}) = \mathbf{b} \otimes \mathbf{a}$. The mode-$n$ product of a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times \cdots I_N}$ and a matrix $\mathbf{B} \in \mathbb{K}^{J_n \times I_n}$ results in the tensor $\mathcal{C} = \mathcal{A} \cdot_n \mathbf{B} \in \mathbb{K}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ with entries $c_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \cdots i_N} b_{j_n i_n}$. It can be seen that $\mathcal{C}$ is equal to a folded version of $\mathbf{B}\mathbf{A}_{(n)}$. A mode-$n$ contraction or tensor–vector product of a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times \cdots \times I_N}$ and a vector $\mathbf{b} \in \mathbb{K}^{I_n}$ is defined as $\mathcal{A} \cdot_n \mathbf{b}^{\mathrm{T}} \in \mathbb{K}^{I_1 \times \cdots I_{n-1} \times I_{n+1} \times \cdots \times I_N}$.

B. Preliminaries on functions

A scalar(-valued) univariate function, e.g., $\underline{f}(u) = 2u$, maps a single scalar variable to another scalar. Multivariate functions depend on several variables which can be scalars, vectors or tensors such as $\underline{f}(u, v) = u + v$ or $\underline{f}(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{K}$. On the other hand, functions can also have vectors in their range rather than scalars, such as $\underline{\mathbf{f}}(u) = \left[1, u, \ldots, u^I\right]^{\mathrm{T}} \in \mathbb{K}^I$. Further generalizations consist of matrix(-valued) and tensor(-valued) functions[2]. Vector, matrix and tensor functions can be seen as the natural (higher-order) stacking of scalar functions. Tensor-valued multivariate functions then merge both extensions, such as the outer product function $\underline{\mathcal{F}}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$.

C. Matrix factorization

Let us consider the following bilinear matrix factorization writing $\mathbf{X} \in \mathbb{K}^{N \times K}$ as a sum of R outer products of nonzero factor vectors:

$$\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r, \tag{1}$$

with $\mathbf{A} \in \mathbb{K}^{N \times R}$ and $\mathbf{B} \in \mathbb{K}^{K \times R}$. The minimal number of rank-1 terms needed to compose $\mathbf{X}$ is called the rank of $\mathbf{X}$.

The decomposition in (1) is not unique as one can find a nonsingular matrix $\mathbf{Z} \in \mathbb{K}^{R \times R}$ such that $\mathbf{A}\mathbf{B}^{\mathrm{T}} = \mathbf{A}\mathbf{Z}\mathbf{Z}^{-1}\mathbf{B}^{\mathrm{T}} = \tilde{\mathbf{A}}\tilde{\mathbf{B}}^{\mathrm{T}}$ with $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{Z}$ and $\tilde{\mathbf{B}} = \mathbf{B}\mathbf{Z}^{-\mathrm{T}}$. *Essential uniqueness* implies that the factor vectors can be uniquely determined up to scaling and permutation. Unless $R = 1$, the decomposition in (1) is only essentially unique if, additionally, constraints on $\mathbf{A}$ and/or $\mathbf{B}$ are imposed. Well-known constraints include orthogonality, triangularity, Vandermonde and nonnegativity, which — whether individually or combined — form the basis of many fundamental factorizations such as the singular value decomposition (SVD), QR decomposition, Vandermonde decomposition and non-negative matrix factorization (NMF).

[2]The terminology should not be confused with the popular class of matrix functions such as the matrix exponential [28]. According to the terminology in this paper, this would be called a matrix-valued function with a matrix-valued variable.

Insert 1 — Component recovery in applications: The bilinear model in (1) forms the basis of various (generic) applications. In some, such as in matrix completion or compression, the unique recovery of the factor vectors is of subordinate importance to the approximation accuracy [29]. In others, this recovery is critical:

- In blind signal separation, for example, the goal is to recover the source signals in $\mathbf{A}$ and/or the mixing matrix $\mathbf{B}$ given only the observed signals in $\mathbf{X}$. Additional assumptions include mutual independence of the source signals, or Vandermonde structure of the mixing matrix when dealing with uniform linear arrays.
- The goal of (hard) clustering consists of the allocation of objects into different groups or clusters [30]. In machine learning, these objects and clusters are typically represented by feature vectors and cluster vectors, respectively. By stacking the feature vectors in the rows of the matrix $\mathbf{X}$ and assuming zero-variance clusters, one can write $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$. The columns of $\mathbf{B}$ consist of the cluster vectors while each row of $\mathbf{A}$ is a canonical basis vector.
- In spectroscopy, $\mathbf{X}$ is an excitation–emission matrix of a chemical mixture. Each factor vector indicates the (scaled) excitation or emission spectrum of a single chemical component.

Each source signal, cluster or chemical component contributes a rank-1 term to the matrix $\mathbf{X}$. These applications, among others, are covered in the remainder of this paper under the generic notion of *unique matrix factorization*. Tensor-based methods seem well suited for this problem due to their interesting properties, as will be discussed.

D. (Coupled) Tensor decompositions

We briefly recall a number of basic concepts from multilinear algebra such as (multilinear) rank and tensor decompositions. For more details, we refer the reader to the provided pointers or to accessible survey papers such as [12]–[14].

1) Rank and decomposition in rank-1 terms: An Nth-order tensor $\mathcal{T}$ has rank one if it can be written as the outer product of N nonzero vectors. The (canonical) polyadic decomposition (CPD) writes $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ as a (minimal) sum of R rank-1 tensors:

$$\begin{aligned} \mathcal{T} &= \sum\nolimits_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \cdots \otimes \mathbf{u}_r^{(N)} \\ &\triangleq \left[\!\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right]. \end{aligned} \tag{2}$$

If each rank-1 term is weighted with a factor c_r, the notation $\left[\!\!\left[\mathbf{c}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right]$ is used.

Generalizing the definition of matrix rank, the rank of $\mathcal{T}$ is defined as the minimal number of rank-1 terms needed to compose $\mathcal{T}$. Unlike in the matrix case, the rank of a tensor can exceed its dimensions. A tensor of size $3 \times 3 \times 3$, for example, can have rank 4.

The CPD in (2) is essentially unique if the factor vectors can be recovered up to permutation of the rank-1 terms, and up to scaling and counterscaling of the factors in the

same rank-1 term. In the tensor case, essential uniqueness is expected under only mild conditions without needing to impose constraints on the factor matrices. These conditions can be deterministic (given a particular set of factor matrices) or generic (assuming factor entries drawn from a continuous probability density function); a concise tutorial is given in [14, Section IV]. Kruskal's condition is a well-known deterministic uniqueness condition and states that the rank-R CPD $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \in \mathbb{K}^{I\times J\times K}$ with $\mathbf{A} \in \mathbb{K}^{I\times R}$, $\mathbf{B} \in \mathbb{K}^{J\times R}$ and $\mathbf{C} \in \mathbb{K}^{K\times R}$ is essentially unique if $k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2$ [9], with the Kruskal rank $k_{\mathbf{M}}$ of a matrix $\mathbf{M}$ defined as the largest integer k such that any k columns of $\mathbf{M}$ are linearly independent. On the other hand, the following is an example of a generic uniqueness condition of a rank-R third-order tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ [31]:[3]

$$R \leq \left\lceil \frac{IJK}{I+J+K-2} \right\rceil - 1 \text{ and } IJK \leq 15000,$$

with $\lceil x \rceil$ denoting the smallest integer not less than x. In the case of $I = J = K = 10$, the rank R can be up to 35 for generic uniqueness. Other deterministic and generic conditions (and order-N generalizations) have been developed in [32]–[36] and [37]–[39], respectively.

This essential uniqueness is only one of the foundations of the success of tensor-based methods and is fundamentally different from essential uniqueness in the matrix case. It allows unique signal separation and a meaningful extraction of features or components, among others. Furthermore, unlike in the matrix case, we have seen that the rank of a tensor can exceed its dimensions. This allows for example the recovery of source signals in an underdetermined setting, i.e., when there are fewer observed signals than source signals. Both aforementioned examples of uniqueness conditions consist of rather mild conditions, meaning that they can result in valid working assumptions in applications.

Let us also briefly consider simultaneous matrix diagonalization. Given a set of K matrices $\mathbf{X}_k \in \mathbb{K}^{I\times J}$, the matrices $\mathbf{A} \in \mathbb{K}^{I\times R}, \mathbf{B} \in \mathbb{K}^{J\times R}$ are then found such that

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{A}\mathbf{\Lambda}_1\mathbf{B}^{\mathrm{T}}, \\ &\vdots \\ \mathbf{X}_K &= \mathbf{A}\mathbf{\Lambda}_K\mathbf{B}^{\mathrm{T}}, \end{aligned}$$

with diagonal matrices $\mathbf{\Lambda}_k \in \mathbb{K}^{R\times R}$. Let $\mathbf{X}_k$ be the kth slice of a tensor $\mathcal{X} \in \mathbb{K}^{I\times J\times K}$. It can be seen that the simultaneous matrix diagonalization problem is equivalent with finding the CPD $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ with the diagonals of the matrices $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_K$ stacked in the rows of $\mathbf{C} \in \mathbb{K}^{K\times R}$. Hence, it is worthwhile to interpret simultaneous matrix diagonalization problems and techniques (such as [15]–[17], [40]) in a tensor-based setting, from which algorithms and uniqueness/identifiability conditions are readily available.

2) Multilinear rank, low multilinear rank decomposition and block term decomposition: The mode-n rank of a tensor $\mathcal{T} \in \mathbb{K}^{I_1\times I_2\times\cdots\times I_N}$ is defined as the rank of the mode-n unfolding $\mathbf{T}_{(n)}$. The multilinear rank of $\mathcal{T}$ is then defined as the tuple of mode-n ranks. Unlike in the matrix case, it can be seen that the mode-n ranks can differ between modes. A multilinear singular value decomposition (MLSVD) writes a multilinear rank-$(R_1, R_2, \ldots, R_N)$ tensor $\mathcal{T}$ as tensor-matrix products of an ordered all-orthogonal core tensor $\mathcal{S} \in \mathbb{C}^{R_1\times R_2\times\cdots\times R_N}$ with orthogonal factor matrices $\mathbf{U}^{(n)} \in \mathbb{K}^{I_n\times R_n}$ for $1 \leq n \leq N$. A tensor with a large theoretical multilinear rank can often be well approximated by a truncated MLSVD. However, for fixed core dimensions R_n, the low multilinear rank approximation (LMLRA) might yield a better approximation [41]. As the core tensor can serve as a compressed version of $\mathcal{T}$, the LMLRA and MLSVD are often used for compression and denoising purposes [42], [43].

3) Block term decomposition, decomposition in multilinear rank-$(L_r, L_r, 1)$ terms and other tools: A block term decomposition (BTD) writes a tensor as a sum of R low multilinear rank terms [44]–[46]. For a third-order tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$, we have

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{G}^{(r)} \bullet_1 \mathbf{A}^{(r)} \bullet_2 \mathbf{B}^{(r)} \bullet_3 \mathbf{C}^{(r)},$$

with $\mathcal{G}^{(r)} \in \mathbb{K}^{L_r\times M_r\times N_R}$, $\mathbf{A}^{(r)} \in \mathbb{K}^{I\times L_r}$, $\mathbf{B}^{(r)} \in \mathbb{K}^{J\times M_r}$ and $\mathbf{C}^{(r)} \in \mathbb{K}^{K\times N_r}$. Each term has multilinear rank-(L_r, M_r, N_r). A special case is the BTD in multilinear rank-$(L_r, L_r, 1)$ terms [47], writing $\mathcal{T}$ as:

$$\mathcal{T} = \sum_{r=1}^{R} \left(\mathbf{A}^{(r)}\mathbf{B}^{(r)^{\mathrm{T}}}\right) \otimes \mathbf{c}_r.$$

These terms are more general than rank-1 terms in a CPD. Nevertheless, it has been shown that the BTD in multilinear rank-$(L_r, L_r, 1)$ terms is still unique under mild conditions without additional assumptions [32], [47].

A decomposition in (rank-L_r $\otimes$ rank-M_r) terms writes a tensor $\mathcal{T}$ as a sum of R terms, each consisting of the outer product of a rank-L_r tensor and a rank-M_r tensor [26].

The tensor train (TT) decomposition represents a higher-order tensor as a set of third-order tensors [48]. It is a particular type of tensor network [49], as well as the hierarchical Tucker (hT) decomposition [50], [51].

4) Tensor decomposition algorithms: The typical algorithms are either algebraic or optimization-based. Examples of the former class build on (generalized) matrix eigenvalue decompositions (GEVD) and can be computed with a polynomial time complexity, e.g., see [36], [52]–[55] for the CPD case. The algebraic algorithms are guaranteed to return exact factors under certain working conditions of an exact decomposition. In the inexact case, e.g., in the presence of noise, they are considered to be suboptimal and can provide good starting values for optimization-based algorithms. Algorithms of the latter class apply alternating optimization techniques, all-at-once methods or stochastic gradient approaches, to name but a few. These algorithms try to minimize the differences between the given tensor and the purported model. Examples for the CPD case are [6], [8], [56] and [57], [58] and [59], respectively. Various toolboxes are available such as Tensorlab

[3]While expected to hold for larger number of entries IJK as well, the condition has only been verified numerically up to 15000 entries.

[60], the N-way toolbox [61] and the Tensor Toolbox [62], among others.

5) Structured and coupled tensor decompositions: Structured tensor decompositions impose additional constraints or structure on the factor matrices, such as orthogonality, nonnegativity or Hankel structure [63]. These constraints can implement application-dependent prior knowledge. Although typically not necessary for essential uniqueness for sufficiently low rank, they can enforce uniqueness even if the rank significantly exceeds the tensor dimensions [64].

Coupled tensor decompositions consider the decomposition of a set of tensors with coupled/shared factor matrices. They provide a necessary tool with respect to the concept of data fusion [63], [65], [66]. For example, given two datasets in the form of third-order tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, the goal could be to find matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(5)}$ such that $\mathcal{T}^{(1)} = [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(5)}]\!]$ and $\mathcal{T}^{(2)} = [\![\mathbf{U}^{(3)}, \mathbf{U}^{(4)}, \mathbf{U}^{(5)}]\!]$ with a coupled factor matrix $\mathbf{U}^{(5)}$ in the third mode. These coupled decompositions are able to provide relaxed uniqueness conditions. Algorithms and uniqueness results can be found in [66]–[68] for coupled CPDs (cCPDs) and in [32], [69] for the coupled decomposition in multilinear rank-$(L_r, L_r, 1)$ terms, among others. A general coupling model considering different types of tensor decompositions can be found in [63].

III. Multiway experiment design

Throughout the years, many powerful tensor tools have been developed to cope with the inherent multiway character of datasets. The acquisition of multiway datasets is an important first step before tensor tools can be applied. Frequently, as will be discussed in this section, experiments readily generate tensor data. In other cases, experiments only yield a single data vector or data matrix. Using tensorization, which will be covered in the remainder of this paper, this vector or matrix data are then transformed to tensors outside of the data creation process. This is illustrated in Figure 2.

Some data obtained from experiments have a natural multiway character. For example, measuring physical quantities such as temperature, volumetric mass density or radiation intensity (think of magnetic resonance imaging (MRI)) along the conventional spatial coordinates readily generates tensor data [70]. A physical quantity of a mixture can also be expressed in a tensor with each mode denoting a component's concentration [71]. As will be discussed in Section IV, such datasets can be seen as evaluated scalar multivariate functions [27], [51], [72], e.g., $\underline{f}(x, y, z)$ for tensors with three spatial modes.

Likewise, the entry at position (i, j, k) of a tensor can indicate, for example, whether subjects i, j and k participated in the same social network discussion [73], giving rise to multiway relational data. Given N objects, each entry of a third-order similarity tensor of dimensions $N \times N \times N$ represents the similarity between three of those objects.

In the previous examples, the modes share a similar meaning. However, this is not necessary. While a grayscale image can be associated with a matrix, a color image can be perceived as tensor data when stacking the coordinates in the color

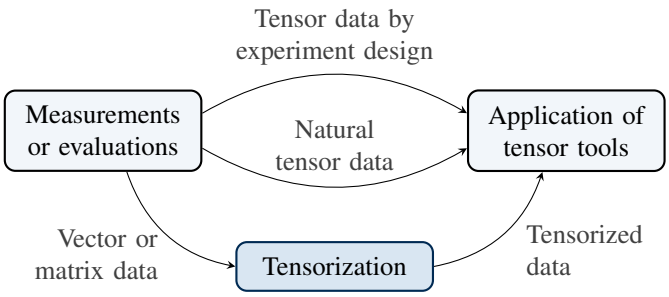


Fig. 2. Measurements and evaluations from experiments or simulations often readily generate tensor data, which can be processed using various tensor tools. If only vector/matrix data are available, one can consider an additional tensorization step.

coding space (such as RGB or CMYK) along a third mode [74], [75]. The first two modes then correspond to the spatial coordinates of the image. Hyperspectral imaging extends color imaging by collecting many more measurements along the third spectral mode [76], [77]. Video data is a natural extension of image data, including a time mode in addition to the pixel and spectral modes [78], [79].

Sometimes one deliberately includes an additional variable in the data creation, allowing for example the creation of tensor data when otherwise only matrix data would be available. One may think of this as tensorization during data acquisition. Incorporating the time variable is a straightforward choice: let an experiment yield matrix data, we can repeat the experiment at various time instances and stack the snapshots in a third-order tensor. Ratings of different users for different movies yield matrix data in movie recommendation systems, while the well-known Netflix dataset additionally includes the rating date yielding a third-order tensor [80]. Other examples can be found, e.g., in social media analysis by taking the time mode into account from e-mail traffic data [73], in food science by collecting flavor release data [81] and in chemometrics by considering the retention time in chromatography [10], [82].

In addition to or besides the temporal diversity, one often exploits the spatial diversity in sensor array processing by providing each receiver with a sensor array rather than a single sensor [83] or in biomedical signal processing by using multiple channels [84], [85]. Experiments can be extended from single-sample or single-subject to multi-sample or multi-subject, e.g., in fluorescence spectroscopy [86], [87], biomedical signal processing [88], face recognition systems [89], [90] and phonetic analysis [6][4]. Tensor faces generalizes standard eigenfaces by incorporating different illuminations, poses and expressions [91]. More generally, one may repeat experiments several times, under varying conditions, with different attributes, etc.

Including additional variables in the measurement process should not be confused with incorporating multiway properties in matrix data. For example, in telecommunications, a unique frequency channel or spreading code is allocated to each source signal. The measured data are strictly speaking

[4]Note that one of the earliest reported multiway datasets can be found in [6].

still matrix data in function of receiver and time. By model construction, one can acknowledge the multiway character and tensorize the matrix data to a third-order tensor using folding or rearranging (also known as segmentation) with the spectral (resp., spreading) diversity resulting in a third mode [92], [93]. This will be further discussed in Section VI-B.

IV. Multivariate and vector-valued functions

Various types of multiway data originate from a single multivariate function. In Section IV-A and IV-B, we discuss the evaluation of scalar and tensor functions, respectively. In Section IV-C, the concept of higher-order derivatives is investigated as a means of transforming scalar functions to tensor functions and, as such, as a tensorization technique. These fundamental notions and results form the basis of many data tensorization techniques discussed in Sections VI and VII.

A. Evaluation of scalar multivariate functions

We make the distinction between scalar-valued variables in Section IV-A1 and vector-valued variables in Section IV-A2.

1) Scalar functions with scalar-valued variables: Let a tensor $\mathcal{F} \in \mathbb{R}^{100\times 100\times 100}$ contain temperature data in various locations in a cuboid-shaped room, with f_{ijk} indicating the temperature at spatial coordinates (x_i, y_j, z_k) for $1 \leq i, j, k \leq 100$. It is obvious that the tensor is an evaluated or discretized form of the scalar continuous function $\underline{f}(x, y, z)$.

More generally, let us consider the evaluation of the function $\underline{f}(u^{(1)}, \ldots, u^{(D)})$ in the Cartesian product grid $\{x_1^{(1)}, \ldots, x_{I_1}^{(1)}\} \times \cdots \times \{x_1^{(D)}, \ldots, x_{I_D}^{(D)}\}$, yielding the order-$D$ tensor $\mathcal{F} \in \mathbb{K}^{I_1\times I_2\times\cdots\times I_D}$ with entries

$$f_{i_1 i_2 \cdots i_D} = \underline{f}(x_{i_1}^{(1)}, x_{i_2}^{(2)}, \ldots, x_{I_D}^{(D)})$$

for all indices [51]. Multiway datasets such as those mentioned in Section III can all be seen as originating from multivariate functions. An alternative notation is $\underline{f}(\boldsymbol{u})$ with $\boldsymbol{u} \in \mathbb{K}^D$. We will use both notations throughout this paper, depending on the context.

Interesting multivariate functions are those that are multiplicative separable in their arguments, i.e., functions that can be written as

$$\underline{f}(u^{(1)}, \ldots, u^{(D)}) = \prod_{d=1}^{D} \underline{g}^{(d)}(u^{(d)}).$$

with univariate functions $\underline{g}^{(d)}(u)$. These functions correspond with a rank-1 tensor $\mathcal{F}$ after evaluation, as $\mathcal{F} = \mathbf{g}^{(1)} \otimes \mathbf{g}^{(2)} \otimes \cdots \otimes \mathbf{g}^{(D)}$ with $g_{i_d}^{(d)} = \underline{g}^{(d)}(x_{i_d}^{(d)})$. An example is the exponential of a linear form:

$$\underline{f}(u^{(1)}, \ldots, u^{(D)}) = \exp(\sum_{d=1}^{D} a_d u^{(d)}) = \prod_{d=1}^{D} \exp(a_d u^{(d)}).$$

Of course, $u^{(d)}$ can be substituted by a general univariate function $\underline{h}(u^{(d)})$, such as $(u^{(d)} - b_i)^2$ in the case of Gaussians.

It follows that if a multivariate function $\underline{f}$ can be written as a sum of R separable functions [94], [95], the evaluated tensor $\mathcal{F}$ admits a rank-R (C)PD. An example is the function $\underline{f} = \sin(\sum_{d=1}^{D} u^{(d)})$ [94]. Using trigonometric identities, it seems that $R = 2^{D-1}$ terms are needed. However, it has been proven using multilinear algebra that $R = D$ is sufficient [96].

2) Scalar functions with vector-valued variables: In a similar way, the scalar function $\underline{f}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)})$ with vector variables $\boldsymbol{u}^{(d)} \in \mathbb{K}^{N_d}$ yields a tensor after evaluation in the Cartesian product $\{\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_{I_1}^{(1)}\} \times \cdots \times \{\mathbf{x}_1^{(D)}, \ldots, \mathbf{x}_{I_D}^{(D)}\}$.

An interesting tensorization technique emerges when the vector variables have the same length, i.e., $N_1, \ldots, N_D = N$. Let us consider a given set of I data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$, and let us stack the evaluations in all possible D-combinations with repetition. An order-D tensor $\mathcal{F} \in \mathbb{K}^{I\times\cdots\times I}$ is then obtained with for all indices

$$f_{i_1 i_2 \cdots i_D} = \underline{f}(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_D}).$$

This is equivalent to the evaluation of $\underline{f}$ in the symmetric Cartesian product $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \cdots \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$. Given only a single set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$, we have effectively constructed a tensor $\mathcal{F}$. As we will see, each tensorization technique from Section VII builds on this mathematical concept.

To illustrate this type of function evaluation, a similarity matrix originates from a set of objects or data vectors (such as feature vectors) and a similarity function $\underline{f}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)})$ (such as the cosine or Gaussian similarity functions), by stacking the different similarities $\underline{f}(\mathbf{x}_i, \mathbf{x}_j)$ between vectors i and j and, hence, by evaluating $\underline{f}$ in the symmetric Cartesian product grid. A higher-order similarity tensor then originates from a function $\underline{f}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)})$.

B. Evaluation of vector- and tensor-valued multivariate functions

It is obvious that a tensor function $\underline{\mathcal{F}}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)}) \in \mathbb{K}^{J_1\times\cdots\times J_E}$ readily yields a tensor $\mathcal{F} \in \mathbb{K}^{J_1\times\cdots\times J_E}$ when evaluating $\underline{\mathcal{F}}$ in a single point, i.e., a single choice of values for $\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)}$. Furthermore, if $I_1 = \cdots = I_D$, one can naturally stack evaluations of $\underline{\mathcal{F}}$ in points of the symmetric Cartesian product grid constructed from a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$. This yields a tensor of order $E + D$ and size $J_1 \times \cdots \times J_E \times I \times \cdots \times I$.

A first special case is the matrix function $\underline{\mathbf{F}}(\boldsymbol{u}) \in \mathbb{K}^{J_1\times J_2}$ which maps a vector $\boldsymbol{u}$ to a matrix and which effectively encompasses a tensorization step, of which a number of examples will be discussed in Section VI. This function generates a tensor $\mathcal{F} \in \mathbb{K}^{J_1\times J_2\times I}$ by stacking the evaluations in a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$.

Second, the vector function $\underline{\mathbf{f}}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}) \in \mathbb{K}^J$ generates a third-order tensor $\mathcal{F}^{J\times I\times I}$ as well, after evaluating $\underline{\mathbf{f}}$ given a set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$. This can also be seen as varying the parameter ρ of a function $\underline{\mathbf{f}}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \rho)$ and stacking the obtained matrices. This will be covered in more detail in Section VII-A.

Insert 2 — Compound matrix: Let us consider the function $\underline{\mathbf{F}}(\mathbf{u}, \mathbf{v}) = \mathbf{u}\mathbf{v}^{\mathrm{T}} - \mathbf{v}\mathbf{u}^{\mathrm{T}} \in \mathbb{K}^{J\times J}$ with $\mathbf{u}, \mathbf{v} \in \mathbb{K}^J$. Given a set of I vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I \in \mathbb{K}^J$, a fourth-order tensor $\mathcal{F}^{J\times J\times I\times I}$ can be constructed by evaluating $\underline{\mathbf{F}}$ in the

Cartesian product grid $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$, with $\mathbf{F}_{kl} = \mathbf{x}_k\mathbf{x}_l^{\mathrm{T}} - \mathbf{x}_l\mathbf{x}_k^{\mathrm{T}}$ for $1 \leq k, l \leq I$.

This tensor is well known in its reduced unfolded form. Indeed, by unfolding $\mathcal{F}$ twice along the first and second, and third and fourth mode, respectively, and by removing the identical rows and columns, one obtains the compound matrix $\mathbf{C}_2(\mathbf{X}) \in \mathbb{K}^{\frac{J(J-1)}{2} \times \frac{I(I-1)}{2}}$ of the matrix $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1, \ldots, \mathbf{x}_I\end{bmatrix} \in \mathbb{K}^{J \times I}$. More general, the Dth compound matrix $\mathbf{C}_D(\mathbf{X})$ can be obtained from order-D tensor function generalizations of $\underline{\mathbf{F}}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)})$.

C. Tensorization using higher-order derivatives

It has been discussed previously that a function $\underline{f}(x, y, z)$ can yield a tensor when evaluated in a product grid. Alternatively, higher-order derivatives have the ability of introducing additional modes and transforming $\underline{f}(x, y, z)$ to a tensor function $\underline{\mathcal{F}}(x, y, z)$ of which the evaluation in a single point results in a data tensor. We make a distinction between scalar and vector functions in Sections IV-C1 and IV-A2. It is curious to see that a series of higher-order derivatives as in a power expansion can be naturally stacked and together yield a single tensor; this will be discussed further below in Section V-B3.

1) Higher-order derivatives of multivariate scalar functions: Let us consider again a multivariate scalar function $\underline{f}(u^{(1)}, \ldots, u^{(D)}) \in \mathbb{K}$. The first-order partial derivatives of $\underline{f}$ form the gradient vector $\nabla \underline{f} \in \mathbb{K}^D$ with $(\nabla \underline{f})_d = \partial \underline{f}/\partial u^{(d)}$ whereas the second-order partial derivatives form the (symmetric) Hessian matrix $\Delta \underline{f} \in \mathbb{K}^{D \times D}$. More general, the order-N partial derivatives can be naturally stacked in a symmetric order-N tensor function $\underline{\mathcal{P}}(u^{(1)}, \ldots, u^{(D)})$ with

$$\left(\underline{\mathcal{P}}(u^{(1)}, \ldots, u^{(D)})\right)_{i_1 i_2 \ldots i_D} = \frac{\partial^N \underline{f}(u^{(1)}, u^{(2)}, \ldots, u^{(D)})}{\partial u^{(i_1)} \partial u^{(i_2)} \cdots \partial u^{(i_D)}}. \tag{3}$$

By evaluating the tensor function in a single point, an order-N tensor $\mathcal{P} \in \mathbb{K}^{D \times D \times \cdots \times D}$ is obtained. While the evaluation of a bivariate function $\underline{f}(u, v)$ in a grid only creates a matrix, a higher-order derivative of $\underline{f}(u, v)$ creates an order-N tensor of dimensions $2 \times 2 \times \cdots \times 2$.

Of particular interest are additive separable scalar functions. Let us assume that the function $\underline{f}$ can be written as a sum of univariate functions, i.e., $\underline{f}(u^{(1)}, \ldots, u^{(D)}) = \sum_{d=1}^{D} g^{(d)}(u^{(d)})$. It can be seen that the higher-order derivatives of additive separable functions are diagonal for $N \geq 2$. Indeed, $\partial^2 \underline{f}/\partial u^{(i)} \partial u^{(j)}$ and corresponding higher-order derivatives are zero if $i \neq j$, and the diagonal entries consist of the unmixed order-N derivatives $\left(\underline{\mathcal{P}}(u^{(1)}, \ldots, u^{(D)})\right)_{ii\ldots i} = \mathrm{d}^N \underline{g}^{(i)}(u^{(i)})/\mathrm{d}\left(u^{(i)}\right)^N$. As will be discussed in Section VII-C2, this diagonality forms the basis of many tensorization-based methods, such as those related to independent component analysis.

2) Higher-order derivatives of multivariate vector functions: Similarly, a multivariate vector function $\underline{\mathbf{f}}(u^{(1)}, \ldots, u^{(D)}) \in \mathbb{K}^J$ (which can be interpreted as a set of scalar functions) can be tensorized using higher-order derivatives. The first-order partial derivatives of $\underline{\mathbf{f}}$ form the Jacobian matrix of size $J \times D$ containing the multivariate scalar functions $\partial \underline{f}_j/\partial u^{(d)}$. Evaluating the Jacobian matrix in a set of K different points and stacking the obtained matrices along the third mode yields a third-order tensor of dimensions $J \times D \times K$.

More general, the order-N partial derivatives fit naturally in a tensor of order $N+1$ with dimensions $J \times D \times \cdots \times D$ that is symmetric in the last D modes.

Again of particular interest are those vector functions consisting of univariate scalar functions which each depend on a different variable. The Jacobian matrix and tensors consisting of the higher-order partial derivatives are then diagonal.

Insert 3 — Decoupling sets of multivariate functions: Consider a set of J multivariate functions $\underline{f}_j(u^{(1)}, \ldots, u^{(D)})$, which together form the vector function $\underline{\mathbf{f}}(u^{(1)}, \ldots, u^{(D)})$. The goal of decoupling the set of multivariate functions, as described in [97], is to obtain a decomposition of the form

$$\underline{f}_j(u^{(1)}, \ldots, u^{(D)}) = \sum_{r=1}^{R} w_{jr} \underline{g}_r \left(\sum_{d=1}^{D} v_{dr} u^{(d)} \right), \tag{4}$$

where $\underline{g}_r(z^{(r)})$ are univariate functions in $z^{(r)} = \sum_{d=1}^{D} v_{dr} u^{(d)}$. The problem has appeared in nonlinear block-oriented system identification [98] and nonlinear state-space identification [99] to simplify complex systems. If the functions $\underline{f}_j$ and $\underline{g}_r$ are polynomials [100], the problem becomes a non-homogeneous multi-polynomial variant of the Waring problem as discussed in Section V-B1.

The unknown matrices $\mathbf{W} \in \mathbb{K}^{J \times R}$ and $\mathbf{V} \in \mathbb{K}^{D \times R}$ (and therefore the functions $\underline{g}_r$) can be recovered by evaluating the Jacobian matrix $\underline{\mathbf{J}}$ in K different points $(x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(D)})$. A tensor $\mathcal{J} \in \mathbb{R}^{J \times D \times K}$ can be constructed by stacking the obtained matrices along the third mode. Let us denote the derivative of $\underline{g}_r(z^{(r)})$ as $\underline{g}'_r(z^{(r)}) = \mathrm{d}\underline{g}_r(z^{(r)})/\mathrm{d}z^{(r)}$. Under the form of (4), each slice of $\mathcal{J}$ can be written as follows:

$$\mathbf{J}_k = \mathbf{W} \mathrm{diag}\left(\underline{g}'_1(z_k^{(1)}), \ldots, \underline{g}'_R(z_k^{(R)}) \right) \mathbf{V}^{\mathrm{T}},$$

with $z_k^{(r)} = \sum_{d=1}^{D} v_{dr} c_k^{(d)}$. Because of the manifesting diagonality and dimensions permitting, $\mathcal{J}$ admits the PD $\mathcal{J} = [\![\mathbf{W}, \mathbf{V}, \mathbf{\Lambda}]\!]$ with $\mathbf{\Lambda} \in \mathbb{R}^{K \times R}$ consisting of entries $\lambda_{kr} = \underline{g}'_r(z_k^{(r)})$ for all indices. Whether $\mathcal{J}$ admits a unique rank-R CPD depends on algebraic conditions, e.g., non-triviality of the evaluation points and full-rankness of $\mathbf{W}$ and $\mathbf{V}$. After estimating $\mathbf{W}$ and $\mathbf{V}$ from the CPD of $\mathcal{T}$, the internal univariate functions $\underline{g}_r(z^{(r)})$ can also be reconstructed. Note that because of the scaling indeterminacy, $\underline{g}_r(z^{(r)})$ is unique up to a scaling of $z^{(r)}$.

V. Tensors representing mathematical objects

Rather than modeling multiway data as discussed in the previous sections, tensors can also represent a mathematical object or operator. Each multilinear function and multivariate polynomial, for example, can be uniquely described by a

tensor, as will be pointed out below. It follows that a number of well-known linear algebra problems and matrix tasks have a clear connection with tensors, such as matrix multiplication and global polynomial optimization.

A. Relations between multilinear functions and tensors

Multilinear functions are multivariate functions that are linear in each variable. Without loss of generality, we restrict the discussion to vector variables for convenience.

1) Multilinear forms: Scalar multilinear functions are also known as multilinear forms. Let us start the discussion with a linear form $\underline{f}$, which maps a vector $\mathbf{u} \in \mathbb{K}^I$ to a scalar. The linearity property implies that $\underline{f}(\alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2) = \alpha_1\underline{f}(\mathbf{u}_1) + \alpha_2\underline{f}(\mathbf{u}_2)$. By writing $\mathbf{u}$ in $\underline{f}(\mathbf{u})$ as $u_1\mathbf{e}_1 + \ldots + u_I\mathbf{e}_I$, $\underline{f}(\mathbf{u})$ equals $u_1\underline{f}(\mathbf{e}_1) + \ldots + u_I\underline{f}(\mathbf{e}_I)$. Collecting $\underline{f}(\mathbf{e}_i)$ in $\mathbf{p} \in \mathbb{K}^I$, it follows that $\underline{f}(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\mathbf{p}$. Hence, each linear form $\underline{f}$ corresponds bijectively to some vector $\mathbf{p}$.

Proceeding further, a bilinear form $\underline{f}$ maps two vectors $\mathbf{u} \in \mathbb{K}^{I_1}$ and $\mathbf{v} \in \mathbb{K}^{I_2}$ to a scalar $\underline{f}(\mathbf{u}, \mathbf{v})$ and is linear in both $\mathbf{u}$ and $\mathbf{v}$. Following the same derivation as before, the bilinear form corresponds to a matrix $\mathbf{P} \in \mathbb{K}^{I_1 \times I_2}$ such that

$$\underline{f}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathrm{T}}\mathbf{P}\mathbf{v} = \mathbf{P} \cdot_1 \mathbf{u}^{\mathrm{T}} \cdot_2 \mathbf{v}^{\mathrm{T}}. \tag{5}$$

More general, a multilinear form $\underline{f}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ with $\mathbf{u}^{(d)} \in \mathbb{K}^{I_d}$ can be represented by and uniquely associated with a mapping tensor $\mathcal{P} \in \mathbb{K}^{I_1 \times \cdots \times I_D}$. Evaluating $\underline{f}$ is then equivalent with computing a number of tensor–vector products, as illustrated in Fig. 3:

$$\begin{aligned}\underline{f}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)}) &= \mathcal{P} \cdot_1 \boldsymbol{u}^{(1)^{\mathrm{T}}} \cdot_2 \boldsymbol{u}^{(2)^{\mathrm{T}}} \cdots \cdot_D \boldsymbol{u}^{(D)^{\mathrm{T}}} \\ &= \sum_{i_1, i_2, \ldots, i_D} p_{i_1 i_2 \cdots i_D} u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(D)}_{i_D}.\end{aligned} \tag{6}$$

If $\mathcal{P}$ has rank 1 and admits $\mathcal{P} = \mathbf{p}^{(1)} \otimes \cdots \otimes \mathbf{p}^{(D)}$, a separation of variables is obtained:

$$\underline{f}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) = \prod_d \left(\mathbf{u}^{(d)^{\mathrm{T}}} \mathbf{p}^{(d)} \right). \tag{7}$$

2) Multilinear vector functions: Let $\underline{\mathbf{f}}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) \in \mathbb{K}^J$ denote a multilinear function which maps D variables to a vector of length J. The function can be seen as a collection of scalar multilinear functions, each corresponding to a specific order-D mapping tensor. By stacking these mapping tensors, it is clear that $\underline{\mathbf{f}}$ corresponds to a tensor $\mathcal{P} \in \mathbb{K}^{J \times I_1 \times \cdots \times I_D}$ of order $D + 1$. An evaluation of $\underline{\mathbf{f}}$ is then equal to a number of tensor–vector products along modes 2 to $D+1$, as illustrated in Fig. 3. The results can be easily extended to tensor functions.

3) Examples: A variety of operators from linear algebra are related to multilinear forms and functions. The trace of a matrix corresponds to a linear form, while the inner product of two vectors is a trivial bilinear form with $\mathbf{P} = \mathbf{I}$. The matrix determinant is linear in each of the columns of the matrix, and hence corresponds to a multilinear form. A special case is the determinant of a 2×2 matrix, which is a bilinear mapping with

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

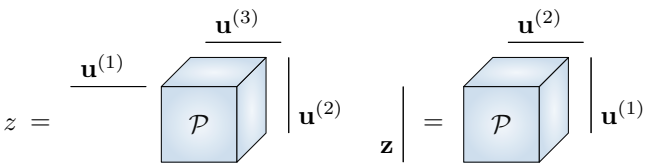


Fig. 3. Illustration of the evaluation of a trilinear form $z = \underline{f}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ (left) and a bilinear vector function $\mathbf{z} = \underline{\mathbf{f}}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ (right).

The matrix multiplication of two matrices and the multiplication of complex numbers each correspond to a bilinear function, as well as the convolution between two vectors.

Insert 4 — Matrix multiplication: Let us consider the matrix multiplication of two square $I \times I$ matrices $\mathbf{A}$ and $\mathbf{B}$, yielding $\mathbf{C} = \mathbf{AB}$. The multiplication maps $\mathbf{A}$ and $\mathbf{B}$ to $\mathbf{C}$, which can be written as $\mathrm{vec}\,(\mathbf{C}) = \underline{f}(\mathrm{vec}\,(\mathbf{A}), \mathrm{vec}\,(\mathbf{B}))$ with the function $\underline{f}$ denoting the multiplication operator. It is a bilinear transformation, as $(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2)\mathbf{B} = \alpha_1(\mathbf{A}_1\mathbf{B}) + \alpha_2(\mathbf{A}_2\mathbf{B})$ and $\mathbf{A}(\beta_1\mathbf{B}_1 + \beta_2\mathbf{B}_2) = \beta_1\mathbf{A}\mathbf{B}_1 + \beta_2\mathbf{A}\mathbf{B}_2$. As discussed, $\underline{f}$ then corresponds to a tensor $\mathcal{P} \in \mathbb{K}^{I^2 \times I^2 \times I^2}$, such that $\mathrm{vec}\,(\mathbf{C}) = \mathcal{P} \cdot_2 \mathrm{vec}\,(\mathbf{A}) \cdot_3 \mathrm{vec}\,(\mathbf{B})$. $\mathcal{P}$ is also called Strassen's multiplication tensor [101]. Compared to the naive matrix multiplication approach, a polyadic decomposition of $\mathcal{P}$ has led to faster algorithms to compute $\mathbf{C}$. It seems that the number of scalar multiplications is determined by the (border) rank of $\mathcal{P}$. In the case of 2×2 matrices, for example, the number of scalar multiplications can be reduced from 8 to 7, the latter being the (border) rank of the corresponding $4 \times 4 \times 4$ multiplication tensor $\mathcal{P}$ [101], [102].

B. Relations between polynomials and tensors

Studies of homogeneous[5] polynomials (also called quantics) in the 19th century actually form the roots of multiway analysis [103], [104]. Homogeneous polynomials can be seen as symmetric versions of multilinear forms, as will be discussed in Section V-B1. Connections between tensors and non-homogeneous polynomials follow in Section V-B2. Theoretical details and further references are provided in [105].

1) Homogeneous multivariate polynomials: Let $\mathbf{A}$ be a symmetric matrix of size $I \times I$. The product of $\mathbf{A}$ with a vector $\mathbf{u} \in \mathbb{K}^I$ along both modes yields a quadratic form:

$$\begin{aligned}\underline{p}(\boldsymbol{u}) &= \boldsymbol{u}^{\mathrm{T}}\mathbf{A}\boldsymbol{u} = \mathbf{A} \cdot_1 \boldsymbol{u}^{\mathrm{T}} \cdot_2 \boldsymbol{u}^{\mathrm{T}} \\ &= \sum_{d=1}^{D} a_{dd} u_d^2 + \sum_{d_1 < d_2}^{D} 2a_{d_1 d_2} u_{d_1} u_{d_2}.\end{aligned} \tag{8}$$

It can be seen that $\underline{p}(\mathbf{u})$ is a degree-2 homogeneous polynomial in I variables. More generally, a homogeneous polynomial $\underline{p}(\mathbf{u})$ of degree D can be associated with a symmetric order-D tensor $\mathcal{A} \in \mathbb{K}^{I \times \cdots \times I}$:

$$\underline{p}(\boldsymbol{u}) = \mathcal{A} \cdot_1 \boldsymbol{u}^{\mathrm{T}} \cdot_2 \boldsymbol{u}^{\mathrm{T}} \cdots \cdot_D \boldsymbol{u}^{\mathrm{T}}. \tag{9}$$

[5]Recall that a homogeneous polynomial contains terms of the same degree.

It can be seen that (8) and (9) are symmetric variants of (5) and (6), respectively. Note that $\underline{p}(\mathbf{u})$ is not linear in $\mathbf{u}$, however.

Insert 5 — The polynomial Waring problem: If $\mathcal{A}$ is a rank-1 tensor in (9), i.e., $\mathcal{A} = \mathbf{b} \otimes \cdots \otimes \mathbf{b}$, the polynomial $\underline{p}(\mathbf{u})$ corresponds to the Dth power of a linear form:

$$\underline{p}(\boldsymbol{u}) = (\mathbf{b} \otimes \cdots \otimes \mathbf{b}) \cdot_1 \boldsymbol{u}^{\mathrm{T}} \cdots \cdot_D \boldsymbol{u}^{\mathrm{T}} = (\boldsymbol{u}^{\mathrm{T}}\mathbf{b})^D .$$

It follows that if $\mathcal{A}$ admits a rank-R CPD, we can write $\underline{p}(\boldsymbol{u}) = \sum_{r=1}^{R} (\boldsymbol{u}^{\mathrm{T}}\mathbf{b}_r)^D$. Finding such a minimal sum of R Dth-power linear forms given a polynomial $\underline{p}(\mathbf{u})$ is also known as the (polynomial) Waring problem [106].

In the quadratic case, the problem can be solved using diagonalization by congruence, using Lagrange's multiplier method or using an eigenvalue decomposition of $\mathbf{A}$ [107], [108]. The general Waring problem for $D > 2$, on the other hand, can not be solved using elementary linear algebra. It is equivalent with finding the symmetric (C)PD $\mathcal{A} = [\![\mathbf{B}, \ldots, \mathbf{B}]\!]$ with $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1, \ldots, \mathbf{b}_R\end{bmatrix}$. The minimal number of terms in the Waring decomposition is equal to the symmetric rank of $\mathcal{A}$ [109], [110].

Insert 6 — From linear to multilinear and polynomial classification: In binary linear classification, one trains or designs a vector $\mathbf{a} \in \mathbb{K}^I$ such that the sign of $\mathbf{f}^{\mathrm{T}}\mathbf{a}$ classifies a given vector $\mathbf{f}$. A (vector) hyperplane then separates two different classes in the I-dimensional space. Multilinear and polynomial classification provide further generalizations. For example, a homogeneous degree-3 polynomial $\underline{p}(\boldsymbol{u}) = \mathcal{A} \cdot_1 \boldsymbol{u}^{\mathrm{T}} \cdot_2 \boldsymbol{u}^{\mathrm{T}} \cdot_3 \boldsymbol{u}^{\mathrm{T}} = (\boldsymbol{u} \otimes \boldsymbol{u} \otimes \boldsymbol{u})^{\mathrm{T}} \mathrm{vec}(\mathcal{A})$ allows the allocation of a feature vector $\mathbf{f}$ to one of both classes by considering the sign of $\underline{p}(\mathbf{f})$. As described in the next section, the homogenization trick allows the extension to affine hyperplanes and non-homogeneous polynomials. One often trains $\mathcal{A}$ under the assumption that it admits a tensor decomposition [111]; the reduced number of variables is welcome for obtaining a meaningful classifier. In support vector machines, non-linear kernel functions are typically used to improve the (multi)linear or polynomial separability, leading to a large value for I. The function $\mathbf{f} \otimes \mathbf{f} \otimes \mathbf{f}$ can actually be seen as a non-linear kernel function by itself.

Insert 7 — Homogeneous polynomial optimization on the unit sphere: Consider a degree-D homogeneous polynomial $\underline{g}(\mathbf{v})$ with $\mathbf{v} \in \mathbb{R}^I$ and with associated order-D symmetric tensor $\mathcal{G} \in \mathbb{R}^{I \times \cdots \times I}$. Recall that $\underline{g}(\mathbf{v}) = \mathcal{G} \cdot_1 \mathbf{v}^{\mathrm{T}} \cdots \cdot_D \mathbf{v}^{\mathrm{T}}$. We want to find the global extremum on the unit sphere (i.e., for $||\mathbf{v}|| = 1$) with the largest absolute value. This is equal to maximizing $(\underline{g}(\mathbf{v}))^2$, which has been shown to be equivalent with minimizing $||\mathcal{G} - [\![\lambda; \mathbf{v}, \ldots, \mathbf{v}]\!]||^2$ on the unit sphere [41]. Hence, the unit sphere optimization corresponds with finding the best (symmetric) rank-1 approximation of $\mathcal{G}$. λ and $\mathbf{v}$ are also known as the largest Z-eigenvalue and corresponding Z-eigenvector of $\mathcal{G}$, respectively [112]–[115].

2) Non-homogeneous multivariate polynomials: A non-homogeneous multivariate polynomial can be characterized by two types of degrees: its total degree, and its coordinate degree. The total degree denotes the maximum degree of each monomial, while the coordinate degree is a tuple consisting of the degrees of the univariate polynomials in each of the variables. For example, a bivariate polynomial with total degree 2 can contain the terms u^2, uv and v^2, while a coordinate degree of $(1, 1)$ allows the term uv but not the terms u^2 and v^2. Each concept has a different connection with tensors.

Starting the discussion with a non-homogeneous polynomial $\underline{p}(\mathbf{u})$ with total degree D and with $\mathbf{u} \in \mathbb{K}^I$, $\underline{p}$ can be seen as a sum of homogeneous degree-d polynomials for $0 \leq d \leq D$:

$$\begin{aligned}\underline{p}(\boldsymbol{u}) = a^{(0)} + \boldsymbol{u}^{\mathrm{T}}\mathbf{a}^{(1)} + \boldsymbol{u}^{\mathrm{T}}\mathbf{A}^{(2)}\boldsymbol{u} + \cdots \\ + \mathcal{A}^{(D)} \cdot_1 \boldsymbol{u}^{\mathrm{T}} \cdots \cdot_D \boldsymbol{u}^{\mathrm{T}},\end{aligned} \tag{10}$$

with $a^{(0)} \in \mathbb{K}$ and with each order-d tensor $\mathcal{A}^{(d)} \in \mathbb{K}^{I \times \cdots \times I}$ corresponding to the degree-d monomial. Let us stack the coefficient tensors symmetrically in a larger order-D tensor $\mathcal{A}^{(\mathrm{hom})} \in \mathbb{K}^{(I+1) \times \cdots \times (I+1)}$, as illustrated in Fig. 4. The non-homogeneous polynomial from (10) can then be written as

$$\underline{p}(\boldsymbol{u}) = \mathcal{A}^{(\mathrm{hom})} \cdot_1 \begin{bmatrix}1, \boldsymbol{u}^{\mathrm{T}}\end{bmatrix} \cdots \cdot_D \begin{bmatrix}1, \boldsymbol{u}^{\mathrm{T}}\end{bmatrix}.$$

The process is called homogenization, and the homogeneous polynomial $\underline{p}^{(\mathrm{hom})}(\boldsymbol{v}) = \mathcal{A}^{(\mathrm{hom})} \cdot_1 \boldsymbol{v}^{\mathrm{T}} \cdots \cdot_D \boldsymbol{v}^{\mathrm{T}}$ is the homogenized form of $\underline{p}(\boldsymbol{u})$, with $\underline{p}^{(\mathrm{hom})}(\begin{bmatrix}1, \boldsymbol{u}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}) = \underline{p}(\boldsymbol{u})$. Hence, a symmetric tensor of size $(I+1) \times \cdots \times (I+1)$ can not only represent a homogeneous polynomial with $I+1$ variables as discussed in Section V-B1, but has a bijective relation with a non-homogeneous polynomial as well. The three inserts from Section V-B1, for example, can then be extended to deal with non-homogeneous polynomials using homogenization.

Let us now consider a non-homogeneous polynomial $\underline{p}(\mathbf{u})$ of coordinate degree $(D_1, \ldots, D_I)$ with $\mathbf{u} \in \mathbb{K}^I$. Unlike in the previous paragraph, $\underline{p}$ can not be uniquely represented by a symmetric tensor $\mathcal{A}$ of size $(I+1) \times \cdots \times (I+1)$, but rather by a tensor $\mathcal{C}$ of size $(D_1 + 1) \times \cdots \times (D_I + 1)$. $\mathcal{C}$ does not need to be symmetric. To see this, let us recall that $\underline{p}(\mathbf{u})$ is a polynomial of degree D_i in the ith variable, and let us define the Vandermonde function $\mathbf{v}(u_i) = \begin{bmatrix}1, u_i, u_i^2, \ldots, u_i^{D_i}\end{bmatrix}^{\mathrm{T}}$. It follows that

$$\underline{p}(\boldsymbol{u}) = \mathcal{C} \cdot_1 \mathbf{v}(u_1)^{\mathrm{T}} \cdots \cdot_I \mathbf{v}(u_I)^{\mathrm{T}}.$$

Observe the connection with the multilinear form of (6), now evaluated in Vandermonde vectors. Second, if $\mathcal{C}$ has rank 1, a product of univariate polynomials is obtained, which is again a separation of variables similar to (7).

Insert 8 — Non-homogeneous polynomial optimization on the real axis across a finite interval: Let $\underline{f}(u)$ be a degree-D non-homogeneous polynomial. The goal consists of finding the global extremum on a finite interval; let us consider $u \in [-1, 1]$ without loss of generality. Broadly stated, $\underline{f}(u)$ can be transformed using a change of variables[6] to a degree-$2D$ homogeneous polynomial $\underline{g}(\mathbf{v}) = \mathcal{G} \cdot_1 \mathbf{v}^{\mathrm{T}} \cdots \cdot_{2D} \mathbf{v}^{\mathrm{T}}$ with $\mathbf{v} \in \mathbb{R}^2$ dependent on u, and with $||\mathbf{v}|| = 1$. Hence, a finite interval on the

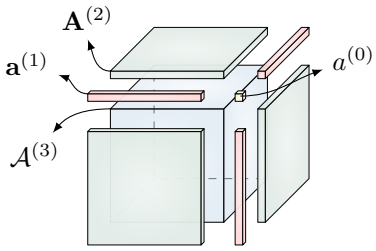


Fig. 4. Illustration of the natural stacking of the coefficient tensors of a cubic (degree-3) multivariate non-homogeneous polynomial with I variables in a tensor $\mathcal{A}^{(\text{hom})}$ with dimensions $(I+1) \times (I+1) \times (I+1)$, which corresponds to the coefficient tensor of a cubic homogenized polynomial with $I+1$ variables. Note that the tensor has been rotated for illustration purposes as $a^{(0)} = a_{111}$.

real axis is mapped to the unit sphere. The entries in $\mathcal{G}$ depend only on the coefficients of $\underline{f}$. Recalling the insert on unit-sphere polynomial optimization of page 9, finding the global finite-interval extremum of a univariate non-homogeneous polynomial is then equivalent with finding the best symmetric rank-1 tensor approximation.

Insert 9 — From systems of linear equations to systems of polynomial equations: It is evident that a system or set of linear equations is associated with the matrix equation $\mathbf{A}\mathbf{u} = \mathbf{b}$. Each row of $\mathbf{A}$ corresponds to a linear form in $\mathbf{u}$. Systems of polynomial equations, on the other hand, are central to kinematic problems from robotics and to various problems in graphical modeling [116]. The following is an example of a bivariate system of polynomials of low degree:

$$\begin{cases} 3u^2 + v - uv = 3 \\ v^3 - u + u^2 v = 5 \end{cases}$$

while Fig. 6 illustrates the solutions of two systems of polynomial equations of moderate degree. As discussed, each polynomial equation can be uniquely represented by a coefficient matrix or tensor. Hence, a system of polynomial equations can be represented by a single tensor by stacking these coefficient matrices or tensors, as illustrated in Fig. 5.

While various procedures are available to find $\mathbf{u}$ in the linear case (from basic row reduction to iterative methods), the polynomial problem is more sophisticated and symbolic methods (using for example Gröbner bases) are often intractable and numerically infeasible [117], [118]. Techniques from numerical linear algebra can be used, e.g., based on the Macaulay null space [119] or the generalized eigenvalue decomposition (GEVD) [120], but there is an increased attention for tensor tools because of the natural multiway representation.

3) Linearization and power expansion: Let us consider a scalar function or signal $\underline{f}$, whether univariate or multivariate.

[6]More specifically, let us parametrize u as $u = \cos t = \cos^2 \frac{t}{2} - \sin^2 \frac{t}{2}$ with $t \in [0, \pi]$. Multiplying each degree-d monomial of $\underline{f}(u)$ with the trivial factor $1 = \left(\cos^2 \frac{t}{2} + \sin^2 \frac{t}{2}\right)^{D-d}$ yields a degree-$2D$ non-homogeneous polynomial in the vector variable $\mathbf{v} = \left[\cos \frac{t}{2}, \sin \frac{t}{2}\right]^{\mathrm{T}}$. Indeed, $||\mathbf{v}|| = 1$.

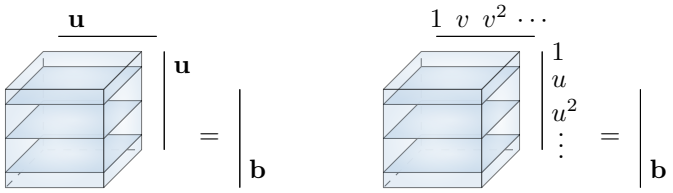


Fig. 5. A representation of systems of polynomial equations by stacking the coefficient matrices along the first mode of a tensor. On the left, a system of multivariate quadratic (total degree 2) polynomials is shown. On the right, a system of bivariate polynomials of fixed coordinate degree is illustrated.

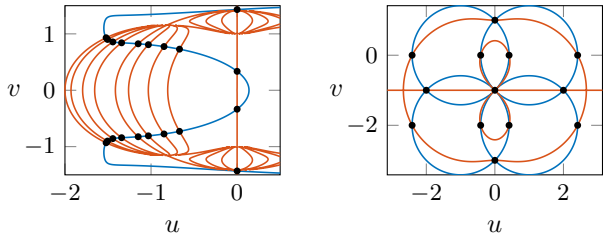


Fig. 6. Two examples of bivariate polynomial systems and their solutions, reproduced from [120].

The process of linearization finds a linear approximation of $\underline{f}$ in a specific point, and aims to provide a good estimate of the function in the neighborhood of this point. It is needless to say that linearization plays a large role in various scientific domains such as microeconomics, optimization theory and control theory regarding dynamical and non-linear systems. Linearization approximates a function $\underline{f}$ in the neighborhood of the point $\mathbf{x}$ then as

$$\underline{f}(\mathbf{u}) \approx \underline{f}(\mathbf{x}) + \mathbf{e}^{\mathrm{T}}\mathbf{a},$$

with $\mathbf{e} = \mathbf{u} - \mathbf{x}$ and with $\mathbf{a} = \nabla \underline{f} \in \mathbb{K}^I$ the gradient of $\underline{f}$.

More general is the concept of approximating $\underline{f}$ with a univariate or multivariate polynomial. Compared to linearization, a broader approximation neighborhood is considered, as illustrated in Fig. 7 for a univariate function. Fundamental is the (truncated) Taylor power expansion, yielding the non-homogeneous polynomial approximation of $\underline{f}$:

$$\begin{aligned} \underline{f}(\mathbf{u}) \approx & \underline{f}(\mathbf{x}) + \mathbf{e}^{\mathrm{T}}\mathbf{a}^{(1)} + \mathbf{e}^{\mathrm{T}}\mathbf{A}^{(2)}\mathbf{e} \\ & + \ldots + \mathcal{A}^{(D)} \cdot_1 \mathbf{e}^{\mathrm{T}} \ldots \cdot_N \mathbf{e}^{\mathrm{T}}. \end{aligned}$$

Each coefficient tensor $\mathcal{A}^{(d)}$ is then equal to $\frac{1}{d!}\mathcal{P}^{(d)}$, with $\mathcal{P}^{(d)}$ the order-d partial derivative tensor of $\underline{f}$ as discussed in Section V-B. Earlier, we have illustrated the stacking of the coefficient tensors of a non-homogeneous polynomial in a single tensor $\mathcal{A}^{(\text{hom})}$ in Fig. 4. Hence, a series of higher-order derivative tensors can also be naturally stacked in a tensor. We can conclude that a polynomial approximation of a multivariate function bijectively corresponds with a tensor.

VI. Tensorization of a single vector

The availability of powerful tensor techniques allows successful results in various applications which can not be straightforwardly obtained using linear algebra. A number

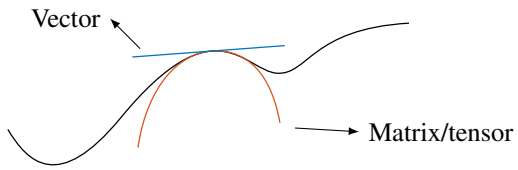


Fig. 7. Illustration of a univariate function or signal approximation in a local area using a linear function, and in a broader area using a quadratic/polynomial function. For multivariate functions, this corresponds with linear algebra and multilinear algebra, respectively.

of examples have been provided in Section III if tensor data are readily available from measurements, observations or evaluations. However, tensor tools can still be applied given vector or matrix data by first tensorizing the data to a tensor.

In this section, we will discuss tensorization techniques given a single vector. This vector can be a time series or an evaluated univariate function, among others. Note that the techniques can also be applied on a matrix by tensorizing each column and by stacking the results. The tensorization step translates the assumed vector model to a tensor model, meaning that the obtained tensor satisfies specific properties for specific types of vectors. We will uncover such relations. Section VII then presents techniques that tensorize a given matrix and that are not suited in general to be applied on a single vector. Although to a much lesser extent, some techniques have appeared in earlier surveys [121], [122].

More mathematically, different matrix-valued functions $\underline{\mathbf{Z}}(\mathbf{f})$ are presented which map or transform a data vector $\mathbf{f}$ to a matrix $\mathbf{Z}$. Furthermore, we present meaningful mappings of a given vector $\mathbf{f}$ to a tensor $\mathcal{Z}$ using tensor-valued functions $\underline{\mathcal{Z}}(\mathbf{f})$. For each method, links are established between the properties of $\mathbf{f}$ and the properties of $\mathbf{Z}$ or $\mathcal{Z}$.

A. Hankel/Toeplitz matrices and tensors

We discuss the strong relationship between Hankel/Toeplitz matrices/tensors on the one hand, and polynomials and exponential functions on the other hand, the latter having a broad relevance for applications such as (multidimensional) harmonic retrieval and direction-of-arrival estimation [33], [83], [123], [124]. We limit the discussion to Hankel matrices/tensors as Toeplitz matrices/tensors are permuted versions; the results can be transferred directly. Unlike in the Toeplitz case, Hankel matrices/tensors offer symmetry in the expressions. Informally, we define *Hankelization* as the process of constructing such Hankel matrices/tensors.

1) Rank-1 and low-rank Hankel matrices: Given a data vector $\mathbf{f} \in \mathbb{K}^N$, a Hankel matrix $\mathbf{H} \in \mathbb{K}^{I \times J}$ with $N = I + J - 1$ is defined as

$$\mathbf{H} = \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & f_J \\ f_2 & f_3 & f_4 & \cdots & f_{J+1} \\ f_3 & f_4 & f_5 & \cdots & f_{J+2} \\ \vdots & \vdots & \vdots & & \vdots \\ f_I & f_{I+1} & f_{I+2} & \cdots & f_N \end{bmatrix}, \tag{11}$$

with entries $h_{ij} = f_{i+j-1}$ for $1 \leq i \leq I$ and $1 \leq j \leq J$. It seems that Hankel matrices can have low rank via two different types of functions: exponentials and polynomials.

First, it is well known that $\mathbf{H}$ has rank 1 if and only if $\mathbf{f}$ is an exponential $\underline{f}(t) = cz^t$ evaluated in an equidistant point set [125]. Given[7] $\mathbf{f} = \begin{bmatrix} c, cz, cz^2, \ldots, cz^{N-1} \end{bmatrix}^{\mathrm{T}}$, we obtain the decomposition with exponential factor vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{H} = c\,\mathbf{a}\mathbf{b}^{\mathrm{T}} \quad \text{with} \quad \begin{cases} \mathbf{a} = \begin{bmatrix} 1, z, z^2, \ldots, z^{I-1} \end{bmatrix}^{\mathrm{T}} \in \mathbb{K}^I, \\ \mathbf{b} = \begin{bmatrix} 1, z, z^2, \ldots, z^{J-1} \end{bmatrix}^{\mathrm{T}} \in \mathbb{K}^J. \end{cases} \tag{12}$$

This result can be readily generalized to sum-of-exponentials. While a single exponential leads to a rank-1 Hankel matrix, a sum of R exponentials with different poles z_r yields a rank-R Hankel matrix $\mathbf{H}$. The latter admits the so-called Vandermonde decomposition with Vandermonde factor matrices $\mathbf{A}$ and $\mathbf{B}$ [126]. Sinusoids and exponentially-damped sinusoids are special cases of sum-of-exponentials as, through Euler's formula, they can be written as a sum of two exponentials.

Second, regarding polynomials, a constant function c leads to a constant rank-1 Hankel matrix[8], while a linear function $\underline{f}(t) = ct$ evaluated in $\{0, 1, \ldots, N-1\}$ leads to a rank-2 Hankel matrix:

$$\mathbf{H} = c \begin{bmatrix} 0 \\ 1 \\ \vdots \\ I-1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + c \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \\ \vdots \\ J-1 \end{bmatrix},$$

as indeed $h_{ij} = c(i+j) = ci + cj$. Recall that $\otimes$ denotes the outer product $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^{\mathrm{T}}$. It follows that a general degree-Q polynomial yields a Hankel matrix of rank $Q + 1$.

Both results form the basis of a large set of low-rank Hankel properties, summarized in Table I. The general class of functions leading to low-rank Hankel matrices can be described by the class of exponential polynomials which are sums and/or products of exponentials, sinusoids and/or polynomials [47]. They can be seen as sum-of-exponentials with poles possibly coinciding in the limit, and can be used to model and approximate a variety of shapes. The well-known (truncated) Fourier series expansion, for example, is a specific case of such an approximation.

2) Hankel tensors: The results can be extended to higher orders in a straightforward way. By Hankelizing each column (or row) of a Hankel matrix again, a third-order Hankel tensor is obtained (or higher order after a number of times). In essence, given a data vector $\mathbf{f} \in \mathbb{K}^N$, we obtain a Hankel tensor $\mathcal{H} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_D}$ with entries $h_{i_1 i_2 \ldots i_D} = f_{i_1 + i_2 + \cdots + i_D - D + 1}$ for all indices and with $N = (\sum_d I_d) - D + 1$ [127]. While Hankel matrices contain anti-diagonal with constant entries, Hankel tensors contain anti-diagonal hyperplanes with constant entries. Note that Hilbert tensors [128] and anti-circulant tensors [129] are specific cases of Hankel tensors.

[7] A sampling in the point set $\{0, 1, \ldots, N-1\}$ is used; nevertheless, the results apply for any set of equidistant points.

[8] Note that a constant function is actually an exponential with $z = 1$.

TABLE I
RANK $r(\mathbf{H})$ OF THE HANKELIZED VERSION OF A DATA VECTOR $\mathbf{f}$ OBTAINED FROM EVALUATING DIFFERENT (EXPONENTIAL) POLYNOMIALS $\underline{f}(t)$ [26]. THE FUNCTION $\underline{p}_r(t)$ IS A POLYNOMIAL OF DEGREE Q_r. THE DIMENSIONS OF $\mathbf{H}$ ARE ASSUMED TO BE SUFFICIENTLY LARGE.

$f(t)$	$r(\mathbf{H})$	$f(t)$	$r(\mathbf{H})$
cz^t	1	$\sum_{r=1}^{R} c_r z_r^t$	R
$c\sin(\omega t+\phi)$ $c\cos(\omega t+\phi)$	2	$\sum_{r=1}^{R} c_r \sin(\omega_r t)$	$2R$
$cz^t \sin(\omega t)$	2	$\sum_{r=1}^{R} c_r z_r^t \sin(\omega_r t)$	$2R$
$p(t)=\sum_{q=0}^{Q} c_q t^q$	$Q+1$	$\sum_{r=1}^{R} p_r(t)$	$\sum_{r=1}^{R} Q_r + R$
$p(t)z^t$	$Q+1$	$\sum_{r=1}^{R} p_r(t) z_r^t$	$\sum_{r=1}^{R} Q_r + R$
$p(t)\sin(\omega t)$	$2Q+2$	$\sum_{r=1}^{R} p_r(t) \sin(\omega_r t)$	$\sum_{r=1}^{R} Q_r + 2R$
$p(t)z^t \sin(\omega t)$	$2Q+2$	$\sum_{r=1}^{R} p_r(t) z_r^t \sin(\omega_r t)$	$\sum_{r=1}^{R} Q_r + 2R$

An exponential $\mathbf{f} = \left[c, cz, \ldots, cz^{N-1}\right]^{\mathrm{T}}$ leads to a rank-1 order-D Hankel tensor with Vandermonde factor vectors:

$$\mathcal{H} = c \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{I_1-1} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{I_2-1} \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{I_D-1} \end{bmatrix} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_D}.$$

More generally, a Hankel tensor constructed from a sum of R exponentials with distinct poles (such as a sum of sinusoids) has rank R. It admits the CPD $\left[\!\left[\mathbf{c}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)}\right]\!\right]$ with Vandermonde matrices $\mathbf{U}^{(d)}$ and scaling vector $\mathbf{c}$, which is a direct generalization of the bilinear Vandermonde decomposition.

The Hankel tensor $\mathcal{H}$ constructed from a degree-Q polynomial, on the other hand, has a more sophisticated structure. It appears that $\mathcal{H}$ has a low multilinear rank $(Q+1, \ldots, Q+1)$ and admits a corresponding LMLRA with a very structured core tensor [47], [126], [130]. It follows that exponential polynomials lead to block term decompositions [47].

Insert 10 — Atomic decomposition: Let $\mathbf{f} = [1, z, \ldots, z^{N-1}]$ be an exponential with N a power of 2, and with $D = \log_2 N$. As discussed, $\mathbf{f}$ yields a rank-1 Hankel matrix/tensor. Let us apply $D-1$ Hankelization steps, resulting in a order-D Hankel tensor $\mathcal{H}$ of size $2 \times 2 \times \cdots \times 2$. Curiously enough, $\mathcal{H}$ admits the following 'atomic' decomposition:

$$\mathcal{H} = \begin{bmatrix} 1 \\ z \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ z \end{bmatrix}.$$

Insert 11 — Pole estimation and harmonic retrieval: Let us consider the problem of estimating the poles of a sum-of-exponentials function, originally solved using Prony's method [131], [132], or using a rotation invariance technique (ESPRIT) [123] and related matrix pencil-based methods [133]. This generic problem appears in many different ways, e.g., in the shape-from-moments problem with the poles indicating the vertices of an unknown to-be-recovered polygon [134], or in signal processing with the poles indicating the frequencies of harmonics [125], [127], [135]. Assuming a model with R poles, the constructed Hankel matrix and higher-order Hankel tensors of a given data vector each have rank R (provided that sufficient samples are available). A low-rank matrix factorization of a Hankel matrix does not readily give a unique solution, while a CPD of a Hankel tensor does. Indeed, under the mentioned uniqueness conditions, the CPD can be unique without Vandermonde constraints. Each factor vector will be (approximately) a single exponential. Adding the Vandermonde constraints can be beneficial though to improve accuracy in a noisy situation, or to enforce uniqueness if the rank significantly exceeds the tensor dimensions [64].

3) Efficient tensor decompositions: Because of the introduced redundancy, the number of entries in a Hankel tensor can significantly exceed the original number of data points. By exploiting the Hankel structure in the tensor decomposition algorithm, however, the tensor does not need to be constructed at all [136], [137] and this so-called curse of dimensionality can be avoided [71]. For instance, a tensor–vector multiplication involving a Hankel tensor of size $I_1 \times I_2 \times \cdots \times I_D$ requires $\mathcal{O}(\prod_{d=1}^{D} I_d)$ flops with a naive implementation. Using fast Fourier transforms, the complexity can be reduced to $\mathcal{O}(N \log N)$ flops with $N = \sum_{d=1}^{D} I_d$ [129], [138].

B. Segmentation, decimation, folding and reshaping

A data vector can be reshaped or folded in two different ways to a matrix. While segmentation extracts and stacks different segments, decimation involves stacking downsampled versions. Furthermore, we give links with Hankelization and data compression.

Given a data vector $\mathbf{f} \in \mathbb{K}^N$, we extract J consecutive segments of length I with $N = IJ$ and stack the segments as the columns of a matrix $\mathbf{S} \in \mathbb{K}^{I \times J}$:

$$\mathbf{S} = \begin{bmatrix} f_1 & f_{I+1} & \cdots & f_{(J-1)I+1} \\ f_2 & f_{I+2} & \cdots & f_{(J-1)I+2} \\ \vdots & \vdots & & \vdots \\ f_I & f_{2I} & \cdots & f_N \end{bmatrix}. \tag{13}$$

For instance, a climate temperature time series can be segmented to a matrix, each row/column corresponding to a particular day/year [139].

Alternatively, one can decimate $\mathbf{f}$ with a certain subsampling factor I and stack the J decimated vectors, resulting in a matrix $\mathbf{D} \in \mathbb{K}^{J \times I}$:

$$\mathbf{D} = \begin{bmatrix} f_1 & f_2 & \cdots & f_I \\ f_{I+1} & f_{I+2} & \cdots & f_{I+1} \\ f_{2I+1} & f_{2I+2} & \cdots & f_{I+2} \\ \vdots & \vdots & & \vdots \\ f_{(J-1)I+1} & f_{(J-1)I+2} & \cdots & f_N \end{bmatrix}.$$

As $\mathbf{D}$ is the transpose of $\mathbf{S}$ in (13), let us continue by only discussing low-rank properties of (variants of) segmentation.

1) Rank-1 matrices: If a vector is constructed from a periodic function with a period equal to I (or a multiple of I), it is straightforward to see that $\mathbf{S}$ has rank 1. Second, an exponential also leads to a rank-1 matrix. Third, a DS-CDMA signal in telecommunications is constructed using spreading codes, which also leads to a rank-1 matrix [92].

More generally, it can be seen that $\mathbf{S}$ has rank 1 and can be written as $\mathbf{S} = \mathbf{a}\mathbf{b}^{\mathrm{T}}$ if and only if $\mathbf{f}$ can be written as the Kronecker product $\mathbf{f} = \mathbf{b} \otimes \mathbf{a}$. Referring back to the examples, a periodic signal leads to a constant vector $\mathbf{b}$, an exponential leads to Vandermonde vectors $\mathbf{a}$ and $\mathbf{b}$, and $\mathbf{b}$ equals the spreading code in DS-CDMA signals.

2) Low-rank matrices and connection with Hankelization: A close link between Hankelization and segmentation emerges. Given $\mathbf{f} \in \mathbb{K}^N$, let us consider the matrices $\mathbf{S} \in \mathbb{K}^{I \times J}$ and $\mathbf{H} \in \mathbb{K}^{I \times (N-I+1)}$ obtained from segmentation and Hankelization, respectively, with $N = IJ$. It seems that the columns of $\mathbf{S}$ are a subset of the columns of $\mathbf{H}$, resulting in $r(\mathbf{S}) \leq r(\mathbf{H})$ [26].

The latter inequality can be understood from the insight that $\mathbf{S}$ offers a more compact representation than $\mathbf{H}$. It follows that the set of data vectors $\mathbf{f}$ leading to a rank-R matrix $\mathbf{H}$ is a subset of those vectors leading to a matrix $\mathbf{S}$ of the same rank. More specifically, the matrix $\mathbf{S}$ has rank R and can be written as $\mathbf{S} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ if and only if $\mathbf{f}$ can be written as the sum of R Kronecker products with $\mathbf{f} = \sum_{r=1}^{R} \mathbf{b}_r \otimes \mathbf{a}_r$ and with full-rank matrices $\mathbf{A} \in \mathbb{K}^{I \times R}$, $\mathbf{B} \in \mathbb{K}^{J \times R}$. Sum-of-exponential functions, polynomials and exponential polynomials from Section VI-A indeed all admit to this condition and give rise to a low-rank matrix $\mathbf{S}$ [26]. Second, if the segment length I is not exactly equal to the period of a periodic signal (e.g., if the period is not exactly known), $\mathbf{S}$ will have a low rank rather than rank 1 [140]. Furthermore, a convolution also leads to a low-rank $\mathbf{S}$.

Generally, a limited amount of Kronecker products can be useful for the approximation of smooth and periodic signals or functions [26]. The low rank property can be exploited to allow various compact representations. This is essential in, e.g., large-scale blind signal separation with a large amount of sensors [26], or in multivariate function approximation where the number of function values exceeds the number of atoms in the universe [27], [50].

3) Tensors obtained from segmentation and decimation: The segmentation (decimation) procedure can be applied again on each column of $\mathbf{S}$ ($\mathbf{D}$). By stacking the obtained matrices along a third mode, a third-order tensor is obtained. This can be applied a number of times to obtain a tensor $\mathcal{S}$ ($\mathcal{D}$) of order D. For example, a time-series of molecule concentrations from multivariate curve resolution can be segmented to a third-order tensor, of which each mode indicates the specific hour, day and year [141], [142].

Again, $\mathcal{S}$ has rank 1 if and only if $\mathbf{f}$ can be written as the Kronecker product of D non-zero vectors, i.e., $\mathbf{f} = \mathbf{u}^{(D)} \otimes \cdots \otimes \mathbf{u}^{(1)}$.

Insert 12 — Quantization: In (very) large-scale applications such as those in scientific computing, a compression or compact representation with a low amount of parameters is often desirable to reduce the computation time. Given a vector $\mathbf{f} \in \mathbb{K}^N$, let us apply $D-1$ segmentation steps to obtain a tensor $\mathcal{S} \in \mathbb{K}^{q \times q \times \cdots \times q}$ of order D, with $N = q^D$. Using a small segment length q and large order D, this process is called *q-adic folding* or *quantization* in tensor-based scientific computing [27], [50], [143], [144]. $\mathcal{S}$ is then approximated using TT decompositions (leading to quantized TT or qTT), hierarchical Tucker decompositions or low-rank CPDs.

Let us focus on the low-rank CPD case, by way of example. The number of degrees of freedom in a rank-R CPD of $\mathcal{S}$ is equal[9] to $(qD - D + 1)R$. A maximal compression of $\mathbf{f}$ is obtained if $q = 2$, leading to only $(1+\log_2 N)R$ degrees of freedom or optimization variables. In the case of small R, this results in a significant reduction compared to the original vector length N. The case of $q = 2$ can be connected with the discussion in Insert 10.

4) Segmentation with overlap and/or with alignment: Hankelization maximally exploits shift invariance, while segmentation offers a maximally compact representation. A compromise between both can be made by selecting a larger subset of columns of the Hankel matrix. This corresponds to the extraction and stacking of overlapping segments of a given vector $\mathbf{f}$.

Another variant consists of the extraction and stacking of non-sequential segments. When tensorizing electrocardiogram (ECG) signals, for example, a specific heartbeat wave indicates the start of each segment [145]–[147].

C. Löwner matrices and tensors

Löwner matrices [148] have an interesting relationship with rational functions, i.e., functions that can be written as a fraction of two polynomials. They frequently occur in the domain of system identification [149], [150]. Remarkably, unlike in the Hankel case, equidistant sampling is not required to obtain low-rank Löwner matrices.

1) Löwner matrix: Consider a vector $\mathbf{f} \in \mathbb{K}^N$, obtained from the evaluation of a function $f(t)$ in the points $\{t_1, \ldots, t_N\}$. Let us construct a matrix $\mathbf{Z} \in \mathbb{K}^{N \times N}$ with $z_{ij} = \frac{f_i - f_j}{t_i - t_j}$. This definition results in undefined fractions[10] on the diagonal of $\mathbf{Z}$, however. Löwner matrices are defined by and constructed from submatrices of $\mathbf{Z}$.

More formally, let us partition[11] the point set $T = \{t_1, \ldots, t_N\}$ in the point sets $X = \{x_1, \ldots, x_I\}$ and $Y = \{y_1, \ldots, y_J\}$, with $N = I + J$. Let vectors $\boldsymbol{\phi} \in \mathbb{N}^I$ and $\boldsymbol{\theta} \in \mathbb{N}^J$ contain the indices of the points from X and Y within T, respectively, e.g., $x_3 = t_{\phi_3}$. The entries of the Löwner

[9]Note that there are qDR variables, while a correction of $(D-1)R$ degrees is introduced regarding the scaling indeterminacy.

[10]In the limit, these diagonal entries would involve the first-order derivatives of $f(t)$ in each of the points $\{t_1, \ldots, t_N\}$.

[11]A partition of a set consists of subsets such that the union of the subsets equals the original set while the intersection of the subsets is empty.

matrix $\mathbf{L} \in \mathbb{K}^{I \times J}$ are then defined as $\frac{f_{\phi_i} - f_{\theta_j}}{x_i - y_j}$ for $1 \leq i \leq I$, $1 \leq j \leq J$, leading to the following matrix:

$$\mathbf{L} = \begin{bmatrix} \frac{f_{\phi_1} - f_{\theta_1}}{x_1 - y_1} & \frac{f_{\phi_1} - f_{\theta_2}}{x_1 - y_2} & \cdots & \frac{f_{\phi_1} - f_{\theta_J}}{x_1 - y_J} \\ \frac{f_{\phi_2} - f_{\theta_1}}{x_2 - y_1} & \frac{f_{\phi_2} - f_{\theta_2}}{x_2 - y_2} & \cdots & \frac{f_{\phi_2} - f_{\theta_J}}{x_2 - y_J} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{f_{\phi_I} - f_{\theta_1}}{x_I - y_1} & \frac{f_{\phi_I} - f_{\theta_2}}{x_I - y_2} & \cdots & \frac{f_{\phi_I} - f_{\theta_J}}{x_I - y_J} \end{bmatrix}. \quad (14)$$

Independent of the type of partitioning, it can be easily seen that $\mathbf{L}$ has rank 1 if[12] $\mathbf{f}$ is a sampled rational function of degree 1. Indeed, if $f(t) = c\frac{t-n}{t-p}$, then $l_{ij} = \alpha \cdot \frac{1}{x_i - p} \cdot \frac{1}{y_j - p}$ with constant $\alpha = -c(p-n)^2$ and we have

$$\mathbf{L} = \alpha\, \mathbf{a}\mathbf{b}^{\mathrm{T}} \quad \text{with} \quad \begin{cases} \mathbf{a} = \left[\frac{1}{x_1 - p}, \ldots, \frac{1}{x_I - p}\right]^{\mathrm{T}} \in \mathbb{K}^I, \\ \mathbf{b} = \left[\frac{1}{y_1 - p}, \ldots, \frac{1}{y_J - p}\right]^{\mathrm{T}} \in \mathbb{K}^J. \end{cases}$$

The factor vectors contain sampled rational functions of degree 1. The expression is similar to the decomposition in (12).

Let us now consider a general rational function of degree[13] R. As a degree-R rational function with distinct poles can be written as a sum of R degree-1 rational functions (its partial fraction decomposition), the corresponding Löwner matrices have rank R [148], [151], [152]. The factor matrices have a Cauchy structure, and the decomposition is also known as the Cauchy decomposition [153]–[156].

Note that polynomials are a special case of rational functions and also lead to low-rank Löwner matrices. More specifically, a degree-Q polynomial yields a rank-Q Löwner matrix.

2) Löwner tensors: To the best of the authors' knowledge, higher-order generalizations of a Löwner matrix have not been defined or investigated in the literature. A Löwner tensor can be obtained by applying the Löwner mapping or *Löwnerization* again along a specific mode of a Löwner matrix, resulting in an order-D tensor $\mathcal{L} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_D}$ with $N = \sum_d I_d$.

This tensor can be defined using a closed form. Consider a partition of T in D sets $X^{(d)} = \left\{x_1^{(d)}, x_2^{(d)}, \ldots, x_{I_d}^{(d)}\right\}$, each associated with an index vector $\phi^{(d)} \in \mathbb{N}^{I_d}$ such that $x_{i_d} = t_{\phi_{i_d}^{(d)}}$. The entries are then defined as follows for all indices:

$$l_{i_1 i_2 \ldots i_D} = \sum_{d=1}^{D} \left[\frac{f_{\phi_{i_d}^{(d)}}}{\prod\limits_{\substack{m=1 \\ m \neq d}}^{D} \left(x_{i_d}^{(d)} - x_{i_m}^{(m)}\right)} \right]. \quad (15)$$

Similar to Löwner matrices, higher-order Löwner tensors also have low (multilinear) rank for rational functions and polynomials. Whereas a Löwner matrix contains first-order finite differences, an order-D Löwner tensor contains finite differences of order $D-1$. It can be seen that (15) simplifies to (14) for $D = 2$, with $l_{ij} = \frac{f_{\phi_i}}{x_i - y_j} + \frac{f_{\theta_j}}{y_j - x_i}$. In [157], it is proven that a degree-R rational function with distinct poles leads to a Löwner tensor of rank R, dimensions permitting.

[12]We would like to point out that this is only a necessary condition and not a sufficient condition. Indeed, only if each possible partitioning of T in sets X and Y leads to a rank-1 Löwner matrix constructed from $\mathbf{f}$, $\mathbf{f}$ is an evaluated rational function of degree 1.

[13]We define the degree of an irreducible rational function as the maximum of the degrees of the polynomial in its numerator and the polynomial in its denominator.

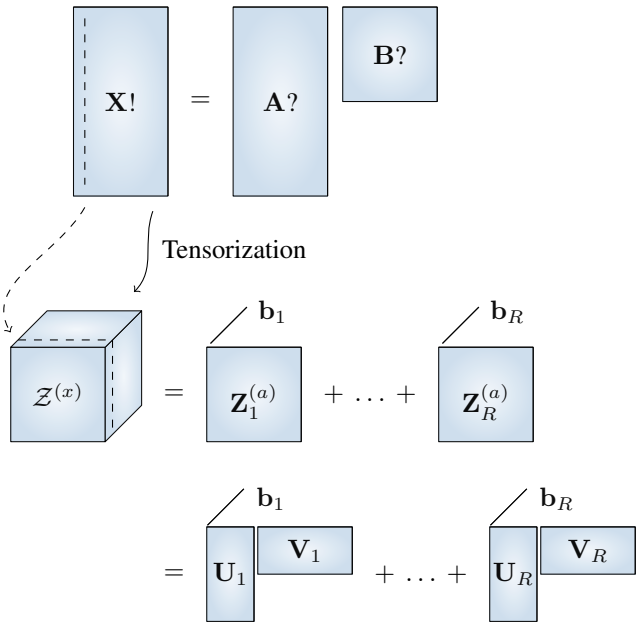


Fig. 8. In deterministic BSS, the matrix $\mathbf{X}$ is mapped to a tensor $\mathcal{Z}^{(x)}$ which is then decomposed in multilinear rank-$(L_r, L_r, 1)$ terms. The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_R$ appear as factor vectors in the third mode.

3) Link with Hankelization: There exists a close link between Löwner matrices and Hankel matrices, with a one-to-one relationship between the class of Löwner matrices of size $I \times J$ and the class of Hankel matrices of the same size. The isomorphism consists of the mapping $\mathbf{H} \mapsto \mathbf{L} = \mathbf{W}_{\mathbf{x}} \mathbf{H} \mathbf{W}_{\mathbf{y}}^{\mathrm{T}}$ with (typically ill-conditioned) matrices $\mathbf{W}_{\mathbf{x}} \in \mathbb{K}^{I \times I}$ and $\mathbf{W}_{\mathbf{y}} \in \mathbb{K}^{J \times J}$ that depend on the point sets X and Y, respectively, and not on the vector $\mathbf{f}$ [153]–[156].

4) Efficient tensor decompositions: Similar to the Hankel case, the structure in Löwner matrices and tensors can be exploited in the case of equidistant points [136], [137]. This avoids the construction of the Löwner matrix/tensor at all.

Insert 13 — Deterministic BSS: The Hankelization, segmentation and Löwnerization methods can be used very effectively for deterministic blind signal separation (BSS) as discussed in [47], [26] and [156], respectively. BSS provides a specific context for unique matrix factorization from Insert 1.

Let us tensorize each vector $\mathbf{x}_k$ (resp. $\mathbf{a}_r$) to a matrix $\mathbf{Z}_k^{(x)} \in \mathbb{K}^{I \times J}$ (resp. $\mathbf{Z}_r^{(a)} \in \mathbb{K}^{I \times J}$), using one of the tensorization techniques. By stacking the matrices, a tensor $\mathcal{Z}^{(x)} \in \mathbb{C}^{I \times J \times K}$ (resp. $\mathcal{Z}^{(a)} \in \mathbb{C}^{I \times J \times R}$) is obtained as illustrated in Figure 8. Let us assume that each matrix $\mathbf{Z}_r^{(a)} = \mathbf{U}_r \mathbf{V}_r^{\mathrm{T}}$ has rank $L_r \leq \min(I, J)$, corresponding to the assumptions on $\mathbf{a}_r$ listed in Table II. As each mapping is linear, we can write:

$$\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \cdot_3 \mathbf{B} = \sum_{r=1}^{R} \mathbf{Z}_r^{(a)} \otimes \mathbf{b}_r = \sum_{r=1}^{R} (\mathbf{U}_r \mathbf{V}_r^{\mathrm{T}}) \otimes \mathbf{b}_r.$$

This boils down to a multilinear rank-$(L_r, L_r, 1)$ decomposition of $\mathcal{Z}^{(x)}$. Under mild conditions on $\mathbf{B}$, $\mathbf{U}_r$ and $\mathbf{V}_r$, which are given in [26], [47], [156], this decomposition is

TABLE II
OVERVIEW OF LOW-RANK CONDITIONS

Tensorization	Low-rank condition
Hankelization	Exponentials, (exponentially-damped) sinusoids, (exponential) polynomials, sum of exponentials
Segmentation	All those from Hankelization, periodic signals, sum of Kronecker products
Löwnerization	Rational functions, polynomials

unique and $\mathbf{B}$ can be directly recovered. An estimate of $\mathbf{A}$ can be obtained from $\mathbf{X}\left(\mathbf{B}^{\dagger}\right)^{\mathrm{T}}$, but also from $\mathcal{Z}^{(a)}$. Only the latter strategy can be used in the underdetermined case. Furthermore, this approach can realize a significant noise reduction. An illustration of deterministic BSS is given in Fig. 9. Note that the tensorization can also be applied on the second mode of $\mathbf{X}$ instead, if the mixing vectors rather than source signals satisfy one of the conditions from Table II. The application of a tensorization technique in the context of (deterministic) BSS together with the use of a decomposition in multilinear rank-$(L_r, L_r, 1)$ terms has been referred to as block component analysis (BCA) [158].

Hankelization-based separation is well known for direction-of-arrival (DOA) estimation with uniform linear arrays (ULA), as the latter correspond to exponential mixing vectors [83]. It has also been applied for the separation of epileptic seizures [159], among others. Segmentation and Löwnerization have both been applied for the separation of maternal and fetal ECG [26], [156], while segmentation is very suitable for large-scale far- and near-field ULA-based DOA estimation [26]. Generalizations to blind system identification have also appeared [160]. Finally, Löwnerization has enabled the separation of amino-acids based on only a single mixture [156]. Typically, excitation–emission spectra of multiple mixtures are required, which are then stacked in a tensor admitting a CPD [86]. By tensorizing a single excitation–emission matrix and acknowledging the fact that emission spectra can be well approximated by low-degree rational functions, a unique separation is obtained as well.

D. Determinant-defining matrix and monomial relations

A relatively new type of matricization exploits monomial relations between signal samples or data vector entries [161]. Various types of signals satisfy such relations, as we will show. We will link monomial relations with rank-deficient matrices and discuss tensor-based approaches for blind signal separation.

1) Monomial relations and rank-deficient matrices: Given a vector $\mathbf{f} \in \mathbb{C}^I$, let us assume there exist indices $p_1, \ldots, p_Q$ and $s_1, \ldots, s_T$, and integer powers $\beta_1, \ldots, \beta_Q$ and $\omega_1, \ldots, \omega_T$, such that

$$f_{p_1}^{\beta_1} f_{p_2}^{\beta_2} \cdots f_{p_Q}^{\beta_Q} = f_{s_1}^{\omega_1} f_{s_2}^{\omega_2} \cdots f_{s_T}^{\omega_T}. \tag{16}$$

Without loss of generality we assume that the degree of the right monomial is equal to or smaller than the degree of the left monomial, i.e., $L = \beta_1 + \ldots + \beta_Q = \omega_0 + \omega_1 + \ldots + \omega_S$ for some $\omega_0 \geq 0$. Let us now construct two vectors:

$$\mathbf{b} = \begin{bmatrix} f_{p_1}\mathbf{1}_{\beta_1} \\ \vdots \\ f_{p_Q}\mathbf{1}_{\beta_Q} \end{bmatrix} \in \mathbb{K}^L \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \mathbf{1}_{\omega_0} \\ f_{s_1}\mathbf{1}_{\omega_1} \\ \vdots \\ f_{s_T}\mathbf{1}_{\omega_Q} \end{bmatrix} \in \mathbb{K}^L.$$

It can be seen that (16) is equivalent with $\prod_{l=1}^{L} b_l = \prod_{l=1}^{L} c_l$, which then leads to the following:

$$\begin{aligned} 0 &= \prod_{l=1}^{L} b_l - \prod_{l=1}^{L} c_l \\ &= b_1 \cdot \begin{vmatrix} b_2 & 0 & \cdots & 0 \\ c_3 & \ddots & \ddots & \vdots \\ & \ddots & \ddots & 0 \\ & & c_L & b_L \end{vmatrix} - c_1 \begin{vmatrix} c_2 & b_2 & & \\ 0 & c_3 & \ddots & \\ \vdots & \ddots & \ddots & b_{L-1} \\ 0 & \cdots & 0 & c_L \end{vmatrix} \\ &= |\mathbf{D}(\mathbf{b}, \mathbf{c})|, \end{aligned} \tag{17}$$

with

$$\mathbf{D}(\mathbf{b}, \mathbf{c}) = \begin{bmatrix} b_1 & 0 & \cdots & 0 & (-1)^L \cdot c_1 \\ c_2 & b_2 & \ddots & & 0 \\ 0 & c_3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & c_L & b_L \end{bmatrix} \in \mathbb{K}^{L \times L}.$$

The first transition is based on the fact that the determinant of a triangular matrix is equal to the product of the diagonal entries. Second, (17) can be seen as the cofactor expansion of $|\mathbf{D}(\mathbf{b}, \mathbf{c})|$ along the first row.

We have effectively matricized the data vector $\mathbf{f}$ to a rank-deficient matrix $\mathbf{D}$. Indeed, (16) leads to $|\mathbf{D}(\mathbf{b}, \mathbf{c})| = 0$. Hence, the rank of $\mathbf{D}$ is strictly smaller than L. On the other hand, the minors in (17) do not vanish except in the trivial case where both sides of (16) are zero. This means that the rank of $\mathbf{D}$ is exactly $L - 1$.

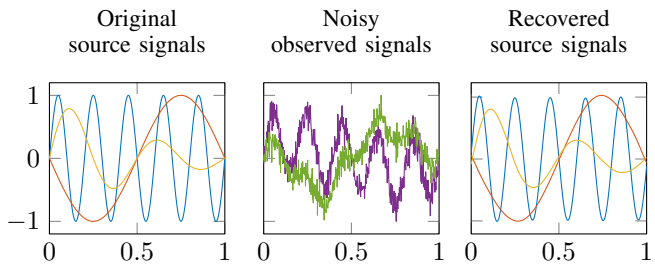


Fig. 9. An example of Hankelization-based deterministic underdetermined BSS. Three signals (shown left, a high-frequency sinusoid (——), a low-frequency sinusoid (——) and an exponentially-damped sinusoid (——)) are mixed together into two signals (middle, —— and ——). A significant amount of white Gaussian noise is furthermore added. Based on only 500 samples of each of the two signals, the goal is to recover the source signals under the assumption that the corresponding Hankel matrices have low rank. On the right, the recovered source signals are shown, using $L_r = 2$ for each source signal. Furthermore, an optimal scaling and permutation step is used to cope with the default BSS indeterminacies. It is remarkable how the signals can still be recovered nearly exactly in this noisy underdetermined situation.

2) Examples: Consider an exponential function $f(t) = cz^t$ sampled in $I = 3$ equidistant points resulting in the data vector $\mathbf{f} \in \mathbb{K}^3$. It can be easily seen that $f_1 f_3 = f_2^2$. We can construct $\mathbf{b} = \begin{bmatrix} f_1, f_3 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{c} = \begin{bmatrix} f_2, f_2 \end{bmatrix}^{\mathrm{T}}$ and the 2×2 matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} b_1 & c_1 \\ c_2 & b_2 \end{bmatrix} = \begin{bmatrix} f_1 & f_2 \\ f_2 & f_3 \end{bmatrix},$$

which is a rank-deficient 2×2 Hankel matrix. A second example involves 4-QAM signals, of which a sample f satisfies the property $f^4 = 1$. Given two 4-QAM samples $\mathbf{f} \in \mathbb{C}^2$, we have $f_1^4 = f_2^4$ and we obtain the following rank-deficient matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} f_1 & 0 & 0 & f_2 \\ f_2 & f_1 & 0 & 0 \\ 0 & f_2 & f_1 & 0 \\ 0 & 0 & f_2 & f_1 \end{bmatrix}.$$

Also BPSK signals, constant modulus and constant power signals satisfy similar monomial relations.

3) Data vectors satisfying multiple monomial relations: The entries of a vector $\mathbf{f} \in \mathbb{K}^N$ can satisfy more than one monomial relation. For example, given an exponential $\mathbf{f}$ with N equidistant samples, $N-2$ relations of the type $f_{n-1} f_{n+1} = f_n^2$ for $n = 2, \ldots, N-1$ are present, among others. Imposing all of these monomial relations corresponds to constraining the 2×2 minors of $\mathbf{H}$ in (11) to be zero, which is equal to the assumption that $\mathbf{H}$ has rank 1.

Second, N samples from a 4-QAM signal satisfy $Z = \frac{I(I-1)}{2}$ relations of the type $f_i^4 = f_j^4$ for $1 \leq i < j \leq N$, yielding Z different rank-deficient matrices of size $L \times L$.

If different types of monomial relations are present in the same data vector, matrices of possibly different size can be constructed. This flexibility can be used when dealing with non-trivial signal processing setups, such as L-shaped and triangular-shaped arrays.

4) Differences with other methods: We have pointed out that given an exponential, Hankelization yields a low-rank matrix while the currently described technique results in a set of rank-deficient matrices. Hankelization, on the one hand, exploits all structure in $\mathbf{f}$ in a rigid manner. On the other hand, monomial relations result in a more flexible approach. They can, for example, be used when an entry is missing, or if a signal is not equidistantly sampled. A link between Löwnerization and the currently described technique can also be established, as, e.g., degree-1 rational functions also admit to specific monomial relations.

Insert 14 — Unique matrix factorization: The usefulness of the monomial relations is discussed in [161] in the context of array processing, harmonic retrieval and blind signal separation. As indicated in Insert 13, ULAs lead to exponential mixing vectors, and constant modulus signals such as 4-QAM and BPSK are often used in telecommunications. In those situations, among others, the mentioned matricization can be used to recover the mixing vectors and/or source signals.

Considering the bilinear factorization in (1), we can assume the columns of $\mathbf{A}$ (or columns of $\mathbf{B}$ for which the results can be transferred easily) to each admit to the same Z monomial relations. Let $\mathbf{D}_{\mathbf{x}_k}^{(z)}$ and $\mathbf{D}_{\mathbf{a}_r}^{(z)}$ be the mapped versions corresponding to the zth monomial relation of $\mathbf{x}_k$ and $\mathbf{a}_r$ with degree L_z, respectively. By stacking the matrices resulting in the tensors $\mathcal{D}_{\mathbf{X}}^{(z)}$ and $\mathcal{D}_{\mathbf{A}}^{(z)}$, the linearity of the mapping allows us to write

$$\mathcal{D}_{\mathbf{X}}^{(z)} = \sum_{r=1}^{R} \mathbf{D}_{\mathbf{a}_r}^{(z)} \otimes \mathbf{b}_r,$$

for $1 \leq z \leq Z$. As each $\mathbf{D}_{\mathbf{a}_r}^{(z)} \in \mathbb{K}^{L_z \times L_z}$ is assumed to have rank $L_z - 1$, the tensors $\mathcal{D}_{\mathbf{X}}^{(z)}$ admit to coupled decompositions in multilinear rank-$(L_z, L_z, 1)$ terms with a coupled factor matrix $\mathbf{B}$. Uniqueness conditions are provided in [161].

E. Time–frequency and time–scale techniques

Time–frequency analysis represents the energy or intensity of a time series as a function of both time and frequency, in contrast to frequency domain representations [23]. It can be well used to investigate non-stationary signals with time-evolving spectral characteristics. We limit the discussion to the short-time Fourier transform and the Wigner–Ville distribution which are treated in Sections VI-E1 and VI-E2, respectively.

An alternative for time–frequency analysis is time–scale analysis. Instead of using an exponential kernel as in the time–frequency case, the signal is analyzed using a wavelet, i.e., a single time-limited wave. The wavelet decomposition, as will be discussed in VI-E3, then compares the given signal to time-delayed and time-scaled versions of this wavelet.

Both time–frequency and time–scale transforms map a signal to a matrix. If multiple signals are available, a tensor can be obtained by stacking the representations of each signal. Space–time–frequency or space–time–scale analysis then considers a set of signals originating from a sensor or antenna array [162]. If a transform is applied along the spatial mode instead of the temporal mode, one refers to space–time–wave analysis [163], [164].

We would like to point out that the time–frequency and time–scale methods have a sparsifying effect for signals that have components that are well-located in both time and frequency or scale. This is the case for a variety of real-life time series and signals. For example, in epileptic encephalogram (EEG) data, epileptic seizures have significantly different spectral components and occur at different time instances compared to standard eye blinking or muscle tension artifacts [159], [165]. The methods can therefore be well used as sparsifying transforms. The sparsity leads to the domains of sparse coding and sparse component analysis (SCA) [166], [167] and has allowed the use of, e.g., the short-time Fourier transform and wavelet decomposition for blind signal separation [168]–[171].

1) Short-time Fourier transform: One way of extracting time-dependent spectral characteristics of a time signal $\underline{f}(t)$ is by considering the spectral contents (Fourier transforms) within small consecutive or overlapping segments of $\underline{f}(t)$. This is also called the Fourier transform with sliding window or

short-time Fourier transform (STFT) [172]. Mathematically, the STFT is defined in the continuous domain as

$$\underline{w}(\omega,\tau)=\int_{-\infty}^{\infty}\underline{f}(t)\underline{h}(t-\tau)e^{-j\omega t}dt,$$

with angular frequency ω and window function $\underline{h}(t)$, commonly a Hann, Hamming or Gaussian[14] window. Note that the squared magnitude of the STFT, i.e., $|\underline{w}(\omega,\tau)|^2$, yields the well-known spectrogram [173].

In the discrete case, there is a clear link between the STFT and the segmentation technique from Section VI-B as different segments are extracted from the signal and then transformed using a discrete Fourier transform (DFT). This can be understood as follows. Let $\mathbf{f}\in\mathbb{K}^N$ and $\mathbf{h}\in\mathbb{K}^L$ be sampled versions of $\underline{f}(t)$ and $\underline{h}(t)$, respectively, using the same equidistant points. The window function $\underline{h}(t)$ is typically designed such that it is approximately zero outside the sampling interval. Let each column of $\mathbf{E}\in\mathbb{K}^{L\times J}$ consist of a length-L segment of $\mathbf{f}$. In the STFT, one typically considers partially overlapping segments rather than consecutive segments. The discrete STFT matrix $\mathbf{W}\in\mathbb{K}^{I\times J}$ corresponding with I uniformly spaced angular frequencies can then be written as

$$\mathbf{W}=\mathbf{Q}\,\mathrm{diag}(\mathbf{h})\mathbf{E},$$

with $\mathbf{Q}\in\mathbb{K}^{I\times L}$ the I-point DFT matrix.

Assuming that $I\geq L$, both $\mathbf{Q}$ and $\mathrm{diag}(\mathbf{h})$ have full column rank, and it follows that the rank of $\mathbf{W}$ is equal[15] to the rank of $\mathbf{E}$. As $\mathbf{E}$ is a segmented version of $\mathbf{f}$, the STFT completely inherits the low-rank properties from the segmentation tensorization technique of Section VI-B, independent of the choice of window function $\underline{h}(t)$. This theoretical connection with segmentation validates the use of low-rank approximations for STFT-based time–frequency representations.

It follows that if a combination of a segmentation and a tensor decomposition is used, such as for the separation of sum-of-exponentials from Insert 13, the segmentation step can be substituted by the STFT, and vice versa. For example, a combination of the STFT and CPD has been used for the separation of EEG data and audio data [174], [175] and audio data [176]. Also other tensor decompositions have been applied in combination with an STFT, e.g., in spreading-multiplexing methods for blind system identification [177]. If segments overlap almost entirely, the STFT is strongly connected with Hankelization following the same rationale.

2) Quadratic representations — Autocorrelation function and Wigner–Ville distribution: Given a zero-mean time signal $\underline{f}(t)$, its non-stationary symmetric autocorrelation function is defined as $\underline{c}(t,\tau)=\underline{f}\left(t+\frac{\tau}{2}\right)\overline{\underline{f}\left(t-\frac{\tau}{2}\right)}$. By applying a Fourier transform on the variable τ, the Wigner–Ville distribution (WVD) is obtained [178], [179]:

$$\underline{w}(\omega,t)=\int_{-\infty}^{\infty}\underline{f}\left(t+\frac{\tau}{2}\right)\overline{\underline{f}\left(t-\frac{\tau}{2}\right)}e^{-j\omega\tau}d\tau,$$

which is a function of the angular frequency ω and lag τ.

[14]The STFT with Gaussian window is also known as the Gabor transform.

[15]Although the rank is equal, the non-zero singular values of $\mathbf{E}$ and $\mathbf{W}$ can differ based on $\mathbf{h}$.

The WVD is not linear in the input signal $\underline{f}(t)$. Indeed, $\underline{w}(\omega,t)$ is a quadratic time–frequency representation. Hence, WVDs can theoretically not be used to solve the case studies described in this section such as pole estimation and blind signal separation from Inserts 11 and 13, as each underlying component does not only contribute a rank-1 term but also contributes additional cross-terms. To diminish the effect of these introduced cross-terms, smoothing kernels are often applied, giving rise to the broad class of Cohen's quadratic time–frequency distributions [23]. A detailed introduction of this class is outside the scope of this paper, but WVDs can be understood as the most simple forms, while the Choi–Williams [180] and Kirkwood–Rihaczek distributions [181], [182] are other well-known quadratic representations, as well as the spectrogram from the previous section.

By considering cross-distributions between signals, the cross-WVD and other quadratic representations can be used meaningfully in a tensor framework as they become linear in each signal. This is discussed in more detail in Section VII-A.

3) Wavelet decomposition: In the STFT from Section VI-E1, the low-frequency and high-frequency components are analyzed using the same fixed segment length. This can limit the spectral analysis if these components appear in respectively broader and narrower parts of the signal, as is the case for various kinds of signals. It might be worthwhile, for example, to consider smaller segments for components with higher frequency so to obtain a better time resolution, i.e., such that high-frequency components can be better distinguished in the time domain. Time–scale analysis offers a solution by comparing the signal at different times with time-scaled (also known as dilated) versions of a wavelet. The continuous wavelet decomposition (CWT) [24], [25], [183], [184] offers such a representation, defined as

$$\underline{w}(\sigma,\tau)=\int_{-\infty}^{\infty}\underline{f}(t)\frac{1}{\sqrt{\sigma}}\overline{\underline{\psi}\left(\frac{t-\tau}{\sigma}\right)}dt, \tag{18}$$

in which the wavelet function $\underline{\psi}(t)$ (such as the Haar or Daubechies wavelet) is similar to the STFT window, but is both translated and dilated. Figure 10 gives a comparison of the time and frequency resolution between the STFT and CWT; note that Heisenberg's uncertainty principle prohibits both superior time and frequency resolution [24].

Given a sampled time signal $\mathbf{f}\in\mathbb{K}^N$, a discretized CWT can readily be computed from (18). Again, a link with segmentation and Hankelization emerges as the discretized CWT $\mathbf{W}\in\mathbb{K}^{I\times J}$ of $\mathbf{f}$ can be written as

$$\mathbf{W}=\mathbf{Z}^{\mathrm{H}}\mathbf{E}, \tag{19}$$

with $\mathbf{Z}\in\mathbb{K}^{M\times I}$ and $\mathbf{E}\in\mathbb{K}^{M\times J}$. Each column vector of $\mathbf{E}$ is a length-M segment of $\mathbf{f}$, while each column vector of $\mathbf{Z}$ is a dilated version of the wavelet $\underline{\psi}(t)$, padded with zeros. The segment length M depends on the largest chosen scale. The formulation distinguishes between the signal component and translation effect on the one hand (captured in $\mathbf{E}$), and the wavelet component and dilation effect on the other hand (captured in $\mathbf{Z}$). Figure 11 illustrates the structure of (19). More technically, each vector $\mathbf{z}_i$ is the finite difference of

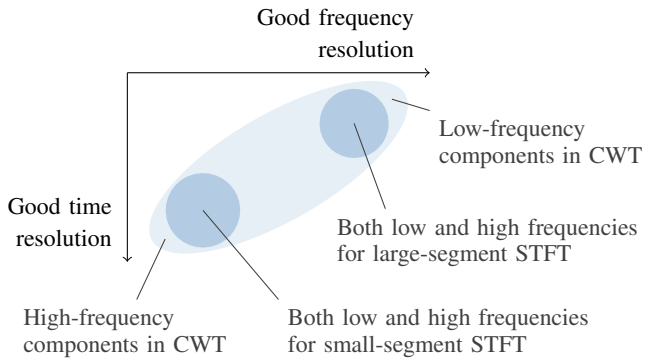


Fig. 10. With small segments, the STFT ( ) offers a good time resolution. It offers a better frequency resolution but inferior time resolution with large segments. The CWT ( ), on the other hand, offers a good time and frequency resolution for high- and low-frequency components, respectively.

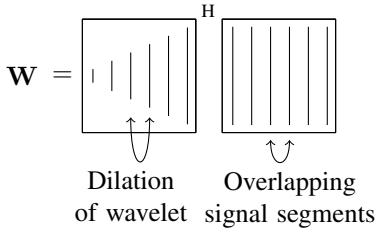


Fig. 11. Illustration of the discrete wavelet decomposition from (19), in which the matrix $\mathbf{Z}$ contains dilated versions of the wavelet $\underline{\psi}(t)$ and the matrix $\mathbf{E}$ contains segmented versions of the given signal.

a decimated version corresponding to the ith scale (possibly padded with zeros) of the function $\underline{\phi}(k) = \int_{-\infty}^{k} \underline{\psi}(t)dt$.

Note that the wavelet transform is linear in the given signal. As $\mathbf{Z}$ has full column rank for $M \geq I$, the rank of $\mathbf{W}$ is theoretically equal to the rank of $\mathbf{E}$, and the discretized CWT inherits the properties from the segmentation and Hankelization techniques. For example, given a sinusoid, the matrix $\mathbf{W}$ has rank 2. Similar to the window function in the STFT, the type of wavelet alters the singular values of $\mathbf{W}$.

Note that in practice, the discrete signal $\mathbf{f}$ can be padded with zeros for the construction of $\mathbf{E}$. While this creates severe artifacts and breaks the low-rank properties, removing a number of columns at the beginning and end of $\mathbf{W}$ suffices to recover the low rank.

The discrete wavelet transform (DWT) provides an alternative for the discretized CWT, using $\underline{\psi}(\frac{t}{\sigma} - \tau)$ rather than $\underline{\psi}\left(\frac{t-\tau}{\sigma}\right)$. It can be seen that this change in variables breaks the low-rank properties of the discretized CWT. On the other hand, a signal that does not yield a low-rank discretized CWT matrix might result in a low-rank matrix after a DWT. We omit a further discussion on the DWT. Note that the DWT formulation is equivalent to discrete filter banks, which will be briefly discussed in Section VI-F.

A choice can be made between different types of wavelets. While we have seen that the choice of wavelet does not alter the theoretical rank, it can influence the sparsifying effect. Different kinds of wavelets have been used in combination with different tensor decompositions. Morlet (also known as Gabor) wavelets, for example, have been used to construct a third-order tensor by stacking the CWTs of different EEG signals, after which a CPD has been applied [174], [185]–[187]. A Ricker or Mexican-hat CWT in combination with a CPD has been used as well [165], [188], while a decomposition in multilinear rank-$(L_r, L_r, 1)$ terms has been applied in, e.g., [159]. Biorthogonal CWTs have appeared in [88], [189] while tensor-based CWT processing has also been applied in hyperspectral imaging [190].

Note that a two-dimensional extension of the wavelet transform can also be applied on matrix data such as image data, possibly using separable two-dimensional wavelets [183]. A third-order tensor is obtained, of which two modes correspond to horizontal and vertical translations while a third mode corresponds to the wavelet dilation. Shearlets are anisotropic multiscale extensions of wavelets, naturally yielding fourth-order tensors [191].

F. Other tensorization techniques

The techniques presented in the Sections VI-A to VI-E are well-known linear mappings from vectors or matrices to tensors with established low-rank properties. There exist a number of other tensorization techniques as well, of which we briefly discuss three techniques. However, low-rank and other meaningful properties validating the use of these techniques in applications are yet to be identified.

A first example consists of Cauchy tensors. A vector $\mathbf{f}$ can be tensorized to an order-D Cauchy tensor $\mathcal{C}$ with entries $c_{i_1 \ldots i_D} = 1/(f_{i_1} + \cdots + f_{i_D})$ [115] or to other types of structured tensors.

Second, the empirical mode decomposition (EMD) extracts components recursively from a given vector $\mathbf{f} \in \mathbb{K}^N$ to obtain a matrix $\mathbf{E} \in \mathbb{K}^{N \times I}$ [192]. Let $\mathbf{e}_i$ be the ith so-called intrinsic mode function (IMF) of $\mathbf{f}$, then $\mathbf{e}_{i+1}$ is the mean of the upper and lower envelope of $\mathbf{e}_i$. An additional Hilbert transform yields the Hilbert–Huang decomposition [193]. Other recursive functions can be applied as well, leading to other mappings.

Uniform tree-structured filter banks form the basis of a tensorization technique as well [194], [195]. The output of the filter bank can be naturally stacked in a tensor, as illustrated in Figure 12. For example, the filters $H_1(z)$ and $H_2(z)$ can then be low-pass and high-pass filters, respectively. Of course, more than two filters can be applied in each stage, the decimators can be omitted and more than two stages can be included. A connection can be made with the DWT from Section VI-E3 [196].

VII. Tensorization of a given set of vectors

The techniques presented in this section consider the availability of a set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$. Provided that each vector has length N, these vectors can be stacked in the columns of a matrix $\mathbf{X}$ of size $N \times K$. Again, these vectors can be time series, evaluated functions or observations from stochastic variables, among others.

On the one hand, we consider the construction of a set of matrices that depend on a parameter. For a chosen set of parameter values, such as lags or frames, a tensor is obtained

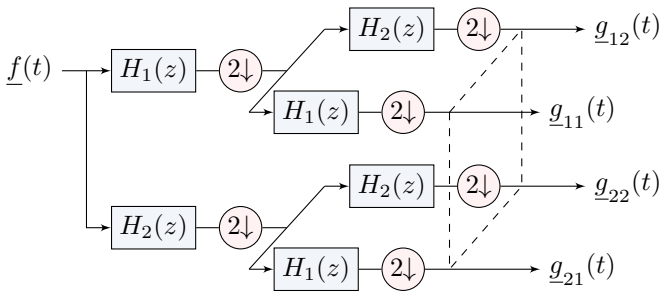


Fig. 12. Uniform tree-structured filter bank in a tensor format.

by stacking the constructed matrices corresponding to each parameter value. Sometimes, as discussed in Section VII-A, the obtained tensor readily admits to a CPD. In other cases, as discussed in Section VII-B, a CPD emerges only after some manipulation. It will become apparent that diagonality is only one of the properties leading to a CPD.

On the other hand, rather than artificially stacking matrices, a tensor with natural multiway character can be directly constructed. Such tensorization techniques can then lead to diagonal tensors such as the context of unique matrix factorization. Higher-order statistics, adjacency tensors and score functions are addressed in Section VII-C.

More mathematically, the techniques in Sections VII-A and VII-B are based on the evaluation of a scalar parameter-dependent multivariate function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ in the Cartesian product grid $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$. A set of parameter values then yields a set of matrices which can readily be stacked in a third-order tensor. Each function $\underline{z}$ can be used in that respect, but is not necessarily meaningful. Similar to Section VI, the links between the vector properties and the properties of the obtained tensor will be discussed. In Section VII-C one then considers a function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ with $D \geq 3$.

A. Collection of matrices by parameter variation

Given $\mathbf{X}$, one can perform a mapping depending on a parameter to generate a set of $K \times K$ matrices $\mathbf{Z}_1, \ldots, \mathbf{Z}_L$. By stacking these matrices, a third-order tensor $\mathcal{Z} \in \mathbb{K}^{K \times K \times L}$ is obtained. For a chosen set of parameter values $\rho_1, \ldots, \rho_L$, the lth matrix $\mathbf{Z}_l$ is then constructed by evaluating a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ as discussed in Section IV-A2, with parameter value ρ_l. Hence, each entry $z_{k_1 k_2 l}$ of $\mathcal{Z}$ is then defined as $\underline{z}(\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \rho_l)$.

We continue by discussing stacked outer products, stacked covariance matrices and stacked Hessian or Jacobian matrices in more detail. Other instances have appeared as well, such as the stacking of time-dependent adjacency matrices. The latter will be covered in Section VII-C3.

Insert 15 — Unique matrix factorization: The techniques from this subsection have been successfully used in the context of unique matrix factorization from Insert 1. The approach then typically consists of the following. Let $\mathbf{Z}_l^{(x)}$ and $\mathbf{Z}_l^{(a)}$ be the parameter-dependent matrices of $\mathbf{X}$ and $\mathbf{A}$, respectively. If $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ is bilinear in $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$, it can be seen that for all entries

$$\begin{aligned} z_{k_1 k_2 l} &= \underline{z}(\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \rho_l) \\ &= \underline{z}(\sum_{r_1} b_{k_1 r_1} \mathbf{a}_{r_1}, \sum_{r_2} b_{k_2 r_2} \mathbf{a}_{r_2}, \rho_l) \\ &= \sum_{r_1, r_2} b_{k_1 r_1} b_{k_2 r_2} \underline{z}(\mathbf{a}_{r_1}, \mathbf{a}_{r_2}, \rho_l) \end{aligned}$$

such that we can write $\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \cdot_1 \mathbf{B} \cdot_2 \mathbf{B}$ or

$$\begin{aligned} \mathbf{Z}_1^{(x)} &= \mathbf{B} \mathbf{Z}_1^{(a)} \mathbf{B}^{\mathrm{T}}, \\ &\vdots \\ \mathbf{Z}_L^{(x)} &= \mathbf{B} \mathbf{Z}_L^{(a)} \mathbf{B}^{\mathrm{T}}. \end{aligned}$$

If the matrices $\mathbf{Z}_l^{(a)}$ can be assumed diagonal given working assumptions on $\mathbf{A}$, the tensor $\mathcal{Z}^{(x)}$ admits a rank-R CPD $\mathcal{Z}^{(x)} = [\![\mathbf{B}, \mathbf{B}, \mathbf{D}^{(a)}]\!]$, as illustrated in Figure 13. The matrix $\mathbf{D}^{(a)}$ consists of the diagonals of $\mathbf{Z}_l^{(a)}$. The CPD is unique under mild conditions on $\mathbf{B}$ and $\mathbf{D}^{(a)}$, which allows the recovery of $\mathbf{B}$ and, subsequently, of $\mathbf{A}$. The function $\underline{z}$ effectively translates the assumptions on $\mathbf{A}$ to diagonal matrix properties and, hence, to low-rank tensor properties.

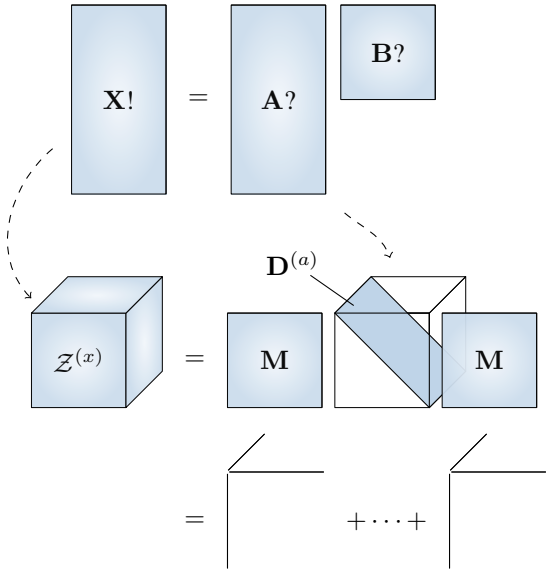


Fig. 13. Because of the bilinearity, the tensor $\mathcal{Z}^{(x)}$ admits $\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \cdot_1 \mathbf{B} \cdot_2 \mathbf{B}$. If the matrices $\mathbf{Z}_l^{(a)}$ are diagonal, a CPD emerges.

1) Stacked outer products: Let us choose the following trivial function for $\underline{z}$ with $\rho \in \{1, \ldots, N\}$:

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho) = u_\rho^{(1)} u_\rho^{(2)}.$$

Hence, if we vary ρ from 1 to N, it can be seen that the nth matrix $\mathbf{Z}_n \mathbb{K}^{K \times K}$ is the symmetric outer product of the nth row of $\mathbf{X}$. The parameter is then the row index. By stacking these outer products, a tensor $\mathcal{Z} \in \mathbb{K}^{K \times K \times N}$ is obtained, with $\mathbf{Z}_{(3)} = \mathbf{X} \odot^{\mathrm{T}} \mathbf{X}$.

Following from the definition, each slice of $\mathcal{Z}$ has rank 1. $\mathcal{Z}$ itself has also rank 1 if and only if $\mathbf{x}_1, \ldots, \mathbf{x}_K$ are scaled versions of each other or, otherwise stated, if $\mathbf{X}$ has rank 1. However, more generally, the rank of $\mathcal{Z}$ is not necessarily equal to the rank of $\mathbf{X}$. This is only true if $\mathbf{X}$ admits to a specific structure, namely if each row vector of $\mathbf{X}$ is a scaled version of one of the columns of a matrix $\mathbf{B} \in \mathbb{K}^{K \times R}$. A specific case appears if each row of $\mathbf{X}$ contains exactly one nonzero entry. Then, and only then, we obtain diagonal[16] matrices $\mathbf{T}_l$. One can think of $\mathbf{X}$ as a (scaled) selection matrix, as the multiplication $\mathbf{X}\mathbf{Y}^{\mathrm{T}}$ 'selects' (and possibly scales) columns from $\mathbf{Y}$.

It is obvious that third-order tensors $\mathcal{Z}_n$ constructed from a generalization of $\underline{z}$ can also be stacked, as well as higher-order generalizations. These generalizations still satisfy the rank and diagonality properties.

Insert 16 — Clustering: By assuming a selection matrix $\mathbf{A}$ in unique matrix factorization, we obtain a clustering model. Indeed, the columns of $\mathbf{B}$ then contain the cluster vectors, and the positions of the nonzero entries in the rows of $\mathbf{A}$ determine the cluster allocations, e.g.:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{b}_4 \end{bmatrix}^{\mathrm{T}}.$$

As the structure in $\mathbf{A}$ allows us to assume that each matrix $\mathbf{F}_n^{(a)}$ is diagonal, $\mathcal{Z}^{(x)}$ admits a rank-R CPD with

$$\mathcal{Z}^{(x)} = \left[\!\!\left[\mathbf{B}, \mathbf{B}, \mathbf{D}^{(a)} \right]\!\!\right]. \tag{20}$$

Furthermore, the diagonal of $\mathbf{Z}_n^{(a)}$ consists of squared entries of the nth row of $\mathbf{A}$. Hence, we have $\mathbf{D}^{(a)} = \mathbf{A} * \mathbf{A}$[17]. The CPD is unique under mild conditions on $\mathbf{A}$ and $\mathbf{B}$, and each cluster contributes a rank-1 term to $\mathcal{T}^{(x)}$. The condition on $\mathbf{A}$, for example, only requires that each column of $\mathbf{A}$ contains at least one nonzero entry, i.e., that from each cluster at least one observation is available.

In topic modeling, for example, each vector $\mathbf{b}_r$ collects the occurrence probability of a set of N dictionary words for the rth topic, and each vector $\mathbf{x}_k$ contains the occurrence frequency of the dictionary words in the kth document [19]. The goal then consists of allocating a topic to each document. It is equal to the 'bag-of-words' problem, and is a specific instance of latent Dirichlet allocation [197]. The latter allows mixed models in which documents can be based on more than one topic.

Instead of occurrence frequency or probability data, the given vectors can also be feature vectors. For example, in MRI processing, stacked outer products have been used to differentiate tumor tissue types, with each tissue type corresponding to a different feature vector and hence different cluster [198].

Note that the CPD formulation avoids the use of heuristic algorithms such as k-means [199]. Even in the presence of noise (i.e., if the clusters have a nonzero variance), algebraic CPD algorithms are still able to give suboptimal estimates of the matrix $\mathbf{A}$ and the cluster vectors in $\mathbf{B}$. If Gaussian noise is added to each row of $\mathbf{X}$ with statistics that depend on the specific cluster the row belongs to, the problem is equivalent to Gaussian mixture parameter estimation [20], [200]. Higher-order moments as discussed in Section VII-C1 may be more appropriate because of their noise reduction abilities.

2) Stacked covariance matrices and other statistics: The vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$ are sometimes observations of K stochastic variables χ_k, collected in the vector $\boldsymbol{\chi} \in \mathbb{K}^K$. The covariance matrix of $\boldsymbol{\chi}$ is then defined as $\mathbf{C}^{(x)} = \mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t)\}$. Given only the observations in $\mathbf{x}_1, \ldots, \mathbf{x}_K$, the sample covariance matrix $\hat{\mathbf{C}}^{(x)} = \frac{1}{N-1}\mathbf{X}^{\mathrm{H}}\mathbf{X}$ is an unbiased estimator of $\mathbf{C}^{(x)}$. The covariance matrix is, among others, a fundamental element of principal component analysis (PCA) [201] as $\mathbf{C}^{(x)}$ is diagonal if and only if the variables χ_k are uncorrelated. Although it is also diagonal for independent variables χ_k, the diagonality is not a sufficient condition for mutual independence. Hence, $\mathbf{C}^{(x)}$ does not provide sufficient information for, e.g., independent component analysis (ICA) [202], [203], such that additional matrices need to be computed. These matrices can then be stacked such that tensor tools can be employed.

One can for example stack different covariance matrices depending on a parameter. Various cases have been reported in the literature. Lagged covariance matrices, for example, compare time-delayed signal versions:

$$\mathbf{C}^{(x)}(\rho) = \mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t-\rho)\}$$

while covariance matrices can also be estimated depending on specific time frames, with s_ϕ and e_ϕ denoting the start and end of the time frame indicated by ρ:

$$\mathbf{C}^{(x)}(\rho) = \mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t)\}_{s_\rho \le t < e_\rho}.$$

Rather than varying only the lag or frame index, both can be varied simultaneously by considering time-frequency representations and selecting specific time–frequency working points, cf. Section VI-E. Indeed, one can for example consider the cross-Wigner–Ville distribution which is closely related to the covariance matrix. The selection of time–frequency points corresponds to the sparsifying effect of the representations [204]. For example, it can be useful to select those time–frequency points where at most one of the underlying signal components is active.

Furthermore, covariance matrices with mutually different complex conjugate conventions can be stacked as well, e.g., $\mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{T}}(t)\}$ and $\mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t)\}$.

[16] Furthermore, each diagonal consists of only a single nonzero entry. More specifically, the diagonal is equal to the element-wise squared corresponding row of $\mathbf{X}$. We ignore this in the remainder.

[17] A more technical way to recognize the CPD is as follows: $\mathbf{Z}_{(3)} = \mathbf{X} \odot^{\mathrm{T}} \mathbf{X} = (\mathbf{A}\mathbf{B}^{\mathrm{T}}) \odot^{\mathrm{T}} (\mathbf{A}\mathbf{B}^{\mathrm{T}}) = (\mathbf{A} \odot^{\mathrm{T}} \mathbf{A})(\mathbf{B}^{\mathrm{T}} \otimes \mathbf{B}^{\mathrm{T}})$. Because of the structure in $\mathbf{A}$, $\mathbf{a}_i * \mathbf{a}_j = \mathbf{0}_N$ for $i \neq j$. By omitting these cross-terms from the expression, we obtain $(\mathbf{A} * \mathbf{A})(\mathbf{B}^{\mathrm{T}} \odot^{\mathrm{T}} \mathbf{B}^{\mathrm{T}}) = (\mathbf{A} * \mathbf{A})(\mathbf{B} \odot \mathbf{B})^{\mathrm{T}}$. The latter is an unfolded version of $\left[\!\!\left[\mathbf{B}, \mathbf{B}, \mathbf{D}^{(a)} \right]\!\!\right]$ along the third mode with $\mathbf{D}^{(a)} = \mathbf{A} * \mathbf{A}$.

Instead of covariance matrices, so-called eigenmatrices of fourth-order cumulants have also been considered. These matrices are defined as reshaped eigenvectors of an unfolded fourth-order cumulant. We will discuss higher-order cumulants in more detail in VII-C2.

Insert 17 — Independent component analysis: The mentioned techniques have been used for ICA, in which one tries to recover a set of mutually independent source signals. Instantaneous ICA is a specific instance of unique matrix factorization from Insert 1, with $\mathbf{X}$ and $\mathbf{A}$ consisting of a set of observed signals and a set of source signals, respectively. Each technique yields diagonal matrices for sets of independent signals such that for each technique a CPD is obtained as discussed in Insert 15.

However, each technique typically requires an additional working assumption. For example, lagged covariance matrices assume that the independent source signals are temporally coherent. These have been applied in the algorithm for multiple unknown signals extraction (AMUSE) [205] with a single lag[18] and in the second-order blind identification (SOBI) algorithm [15], [206] with multiple lags. Time-depending covariance matrices, on the other hand, assume non-stationary signals. These have not only been applied in a signal processing context [207], [208] but also in a context of directed network topology inference [209]. Stacking covariance matrices with mutually different complex conjugate conventions enables the separation of non-circular signals [210]. Furthermore, by working with matrices evaluated in different time–frequency points, sparsity is typically assumed [168], [169], [211]. Finally, the eigenmatrices of fourth-order cumulants assume non-Gaussian source signals.

Note that together with each technique-dependent working assumption, there are typically mild algebraic conditions present to allow uniqueness of the CPD of $\mathcal{Z}^{(x)}$. In SOBI, for example, it is required that the autocorrelations of the different source signals differ [15].

3) Stacked Hessian or Jacobian matrices: A tensor can be obtained by stacking Hessian or Jacobian matrices evaluated in different points, as discussed in Section IV-C. An example is the covariance matrix which is the Hessian matrix of the second characteristic function evaluated in the origin. Hence, the techniques from Section VII-A2 all fall within the class of stacked Hessian matrices. Note that the origin is merely a practical choice; other evaluation points can also be used but lead to more advanced expressions [214], [215].

Instead of Hessian matrices, one can also stack Jacobian matrices. This has been used, e.g., in the context of multivariate function decoupling, as discussed in Insert 3.

One might wonder how the concept of derivatives is related to the evaluation of a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ as discussed in the beginning of this section. The answer is not straightforward, and will be provided in Section VII.

[18]There is a very close link as well between AMUSE and canonical correlation analysis [212], [213].

B. Collection of matrices indirectly resulting in a CPD

In the previous Section VII-A, a number of techniques have been reviewed which stack parameter-depending matrices. In various cases, these techniques have enabled the use of a CPD through the fact that the constructed matrices are diagonal for a specific set of vectors. For example, stacked outer products yield diagonal matrices for a given selection matrix, while covariance matrices are diagonal for independent signals. This diagonal property readily yields a CPD.

However, algorithms have appeared in which a CPD is obtained only after some manipulations. In this section, we will show that these algorithms can each be linked to the construction of a tensor by stacking different matrices. Instead of a diagonal property for a specific set of vectors, these matrices satisfy other kinds of properties.

Unlike in other sections, we will focus specifically on the problem of unique matrix factorization, i.e., the recovery of $\mathbf{A}$ and $\mathbf{B}$ given $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$. In the previous CPD appearances in applications such as deterministic BSS from Insert 13, clustering from Insert 16 and ICA from Insert 17, one or more of the factor matrices was equal to $\mathbf{B}$. In this section, it can be seen that not $\mathbf{B}$ is recovered from the CPD but rather a so-called 'separation matrix' $\mathbf{W} = \left(\mathbf{B}^{-1}\right)^{\mathrm{T}}$ such that $\mathbf{X}\mathbf{W} = \mathbf{A}$.

We start the discussion with an introductory case study on the separation of constant modulus signals in Section VII-B1. We then generalize the procedure in Section VII-B2 and provide an overview of the properties that also lead to a CPD after some manipulations, apart from the diagonal property. We then provide a number of examples of the technique in Section VII-B3.

1) Case study – Separation of constant modulus signals: Let us assume that each entry of $\mathbf{A}$ has a constant modulus, satisfying the condition $|a_{nr}|^2 = a_{nr}\overline{a}_{nr} = c$, with $c \in \mathbb{R}$ the constant squared modulus. Each vector $\mathbf{x}_k$ is then a mixture of the constant modulus signals $\mathbf{a}_1, \ldots, \mathbf{a}_R$ [17]. We assume $K = R$ but we continue to use both variables to improve readability. If $K > R$ we can use a dimension reduction technique to obtain $K = R$ by using for example PCA [216].

Similar to Section VII-A, we will apply a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$, namely

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho) = u_\rho^{(1)} \overline{\left(u_\rho^{(2)}\right)}.$$

The parameter ρ corresponds with the sample index n. After evaluation in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ and $\{\mathbf{a}_1, \ldots, \mathbf{a}_R\} \times \{\mathbf{a}_1, \ldots, \mathbf{a}_R\}$ for $n = 1, \ldots, N$, we obtain the tensors $\mathcal{Z}^{(x)} \in \mathbb{K}^{K \times K \times N}$ and $\mathcal{Z}^{(a)} \in \mathbb{K}^{R \times R \times N}$ with slices $\mathbf{Z}_n^{(x)}$ and $\mathbf{Z}_n^{(a)}$, respectively. The function $\underline{z}$ is linear in both $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ such that, as discussed in Insert 15, we can write

$$\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \bullet_1 \mathbf{B} \bullet_2 \overline{\mathbf{B}}. \tag{21}$$

or similarly

$$\begin{aligned} \mathbf{Z}_1^{(x)} &= \mathbf{B}\mathbf{Z}_1^{(a)}\mathbf{B}^{\mathrm{T}}, \\ &\vdots \\ \mathbf{Z}_N^{(x)} &= \mathbf{B}\mathbf{Z}_N^{(a)}\mathbf{B}^{\mathrm{T}}. \end{aligned}$$

In this case, unlike in Section VII-A and Figure 13, the matrices $\mathbf{Z}_n^{(a)}$ are not diagonal. However, we know that each diagonal entry of $\mathbf{F}_n^{(a)}$ is constant and equal to c. Under this constraint, the problem in (21) can be solved in a classical optimization setting. On the other hand, it is clear that there is only little difference with solving the standard problem $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ under the constant modulus constraint on $\mathbf{A}$. However, as we will further show, (21) does provide a basis to derive algebraic algorithms and uniqueness conditions.

Let us consider the previously defined separation matrix $\mathbf{W} \in \mathbb{K}^{K\times R}$ such that $\mathbf{A} = \mathbf{X}\mathbf{W}$. For each separation vector $\mathbf{w}_r$, we obtain a constant modulus vector $\mathbf{X}\mathbf{w}_r = \mathbf{a}_r$. Following from (21), we can write

$$\mathcal{Z}^{(x)} \bullet_1 \mathbf{W}^{\mathrm{T}} \bullet_2 \mathbf{W}^{\mathrm{H}} = \mathcal{Z}^{(a)}.$$

which is equivalent to

$$\mathbf{Z}_{(3)}^{(x)} \left(\overline{\mathbf{W}} \otimes \mathbf{W}\right) = \mathbf{Z}_{(3)}^{(a)}. \tag{22}$$

Let us only consider those equations corresponding to the diagonals of $\mathbf{Z}_n^{(x)}$ and $\mathbf{Z}_n^{(a)}$, and let $\mathbf{D}^{(a)}$ again contain the diagonals of $\mathbf{Z}_n^{(a)}$ as in Section VII-A, i.e., $\mathbf{D}^{(a)} = \left[\mathrm{diag}(\mathbf{Z}_1^{(a)}), \ldots, \mathrm{diag}(\mathbf{Z}_N^{(a)})\right]^{\mathrm{T}} \in \mathbb{C}^{N\times R}$. (22) then reduces to

$$\mathbf{Z}_{(3)}^{(x)}\mathbf{Y} = \mathbf{D}^{(a)} = c\mathbf{1}_{N\times R}, \qquad \text{with } \mathbf{Y} = \overline{\mathbf{W}} \odot \mathbf{W}. \tag{23}$$

This is a linear system in which each solution has a Kronecker structure, i.e., $\overline{\mathbf{w}}_r \odot \mathbf{w}_r$. This can be solved with suitable algebraic or optimization-based algorithms.

Let us go one step further and relax the structure of the columns of $\mathbf{Y}$ to reduce (23) to a homogeneous system. This will allow us to uncover a CPD in the null space of $\mathbf{Z}_{(3)}^{(x)}$ from which $\mathbf{W}$ can be recovered. By applying a Householder or discrete Fourier transform, among others, and omitting the first row, the problem can be easily transformed to

$$\tilde{\mathbf{Z}}_{(3)}^{(x)}\mathbf{Y} = \mathbf{0}_{N\times R} \tag{24}$$

with $\tilde{\mathbf{Z}}_{(3)}^{(x)} \in \mathbb{C}^{(N-1)\times K^2}$. Given (24), we know that the dimension of the kernel of $\tilde{\mathbf{Z}}_{(3)}^{(x)}$ is at least R. This corresponds with the fact that there are R separation vectors yielding R constant modulus signals. Let us assume that there are sufficient samples available such that the number of rows of $\tilde{\mathbf{Z}}_{(3)}^{(x)}$ is larger than $K^2 - R$, and that the kernel dimension is exactly equal[19] to R. Let the columns of $\mathbf{N} \in \mathbb{C}^{K^2\times R}$ form a basis of this null space. There exists a matrix $\mathbf{\Lambda} \in \mathbb{C}^{R\times R}$ such that $\mathbf{N} = \mathbf{Y}\mathbf{\Lambda}^{\mathrm{T}} = \left(\overline{\mathbf{W}} \odot \mathbf{W}\right)\mathbf{\Lambda}^{\mathrm{T}}$. Hence, by reshaping $\mathbf{N}$ to the tensor $\mathcal{N} \in \mathbb{C}^{K\times K\times R}$, the rank-$R$ CPD $\mathcal{N} = \left[\!\left[\mathbf{W}, \overline{\mathbf{W}}, \mathbf{\Lambda}\right]\!\right]$ is obtained. After estimating $\mathbf{W}$, both $\mathbf{A}$ and $\mathbf{B}$ can be recovered. This sequence of steps forms the basis of the well-known analytical constant modulus algorithm (ACMA) [17]. Note that this third method is fundamentally different from the second method, as solving (23) on the one hand and computing the CPD of $\mathcal{N}$ after the computation of the null space from (24) on the other hand both involve different weightings.

[19]This corresponds to $r(\tilde{\mathbf{Z}}_{(3)}^{(x)}) = K^2 - R$, which is a valid assumption if the phases of the source samples are sufficiently random, cf. [17], [217].

2) Generalization of the concept: The case study from the previous section leads us to an important novel concept: as long as a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ can be found that yields a tensor $\mathcal{F}^{(a)}$ of which the slice diagonals consist of constant entries, a CPD emerges provided a number of algebraic conditions are satisfied. However, the concept can be further generalized as we will discuss now. A CPD can also be obtained if the diagonals of $\mathbf{F}_l^{(a)}$ (cf. Figure 13) yield a zero matrix, a rank-1 matrix, or more generally, a matrix with rank at most $R-2$.

Let us first discuss the rank-1 condition. A Householder or discrete Fourier transform is not necessary to uncover a CPD from (23). It can be seen that the constant matrix $c\mathbf{1}_{N\times R} = c\mathbf{1}_N\mathbf{1}_R^{\mathrm{T}}$ has rank 1; let us assume that $\mathbf{D}^{(a)} = \mathbf{g}\mathbf{h}^{\mathrm{T}}$ with $\mathbf{g} \in \mathbb{K}^N$ and $\mathbf{h} \in \mathbb{K}^R$. Hence, under the same conditions as before regarding the number of samples and the column rank, it can be assumed from (23) that the kernel of $\mathbf{Z}_{(3)}^{(x)}$ has exactly dimension $R-1$. Let the columns of $\mathbf{N} \in \mathbb{K}^{K^2\times(R-1)}$ form a basis of this kernel. By folding $\mathbf{N}$ to a tensor $\mathcal{N} \in \mathbb{K}^{K\times K\times(R-1)}$ as before, a rank-R CPD $\mathcal{N} = \left[\!\left[\mathbf{W}, \overline{\mathbf{W}}, \mathbf{\Lambda}\right]\!\right]$ emerges again. Note that we do not need to know the values in $\mathbf{g}$ or $\mathbf{h}$ to obtain the CPD.

The rank-1 condition on $\mathbf{D}^{(a)}$ can be relaxed even further, as the rank of $\mathbf{D}^{(a)}$ can be up to $R-2$. In the latter case, as long as the algebraic conditions are satisfied, a tensor $\mathcal{N} \in \mathbb{K}^{R\times R\times 2}$ is obtained that admits to a rank-R CPD. Although $\mathcal{N}$ only contains 2 slices, the CPD can still be unique following from Kruskal's uniqueness conditions [9]. Note that the rank of $\mathbf{D}^{(a)}$ can not exceed further than $R-2$. If $r(\mathbf{D}^{(a)}) = R-1$, for example, the null space basis consists of a single vector and folding this vector only yields a matrix rather than a third-order tensor which does not provide sufficient information to obtain a unique solution.

On the other hand, instead of having rank 1 or higher rank, $\mathbf{D}^{(a)}$ can also immediately be a zero matrix, i.e., $\mathbf{D}^{(a)} = \mathbf{0}_{N\times R}$. A reduction to a homogeneous system as in (24) is then obviously not necessary. Folding a basis of the null space of $\mathbf{Z}_{(3)}^{(x)}$ readily yields a rank-R tensor $\mathcal{N} \in \mathbb{C}^{K\times K\times R}$ with R slices, provided the same conditions as before are satisfied. Note that $\mathbf{D}^{(a)} = \mathbf{0}_{N\times R}$ results in hollow matrices $\mathbf{Z}_n^{(a)}$, i.e., the diagonals of $\mathbf{Z}_n^{(a)}$ are zero. It is interesting to connect this to the techniques from Section VII-A such as SOBI. There, the diagonal entries of $\mathbf{Z}_l^{(a)}$ were assumed to be non-zero while the off-diagonal entries were assumed to be zero. Here, it is exactly the opposite. As we have uncovered, both properties effectively lead to a CPD, whether directly or indirectly.

There is an important connection with the technique from Section VI-D. While the latter considers monomial relations, the function $\underline{z}$ in this section can implement the broader class of polynomial relations. The vectors $\mathbf{a}_r$ are then assumed to lie on algebraic varieties. If $\mathbf{D}^{(a)} = \mathbf{0}_{N\times R}$, for example, the algebraic variety is then the set of solutions of a system of homogeneous polynomial equations.

3) Other examples: One of the first appearances of the technique can be found in [218], discussing the decomposition of the fourth-order cumulant[20] in the context of underdetermined

[20]See Section VII-C2

ICA. The presented so-called rank-1 detection device was later applied to obtain an algebraic algorithm for a CPD of which only one factor matrix has full column rank [54]. Indeed, the CPD $\mathcal{X} = [\![\mathbf{G}, \mathbf{H}, \mathbf{B}]\!]$ with $\mathbf{G} \in \mathbb{K}^{I \times R}$ and $\mathbf{H} \in \mathbb{K}^{J \times R}$ can be seen as a unique matrix factorization problem $\left(\mathbf{X}_{(3)}\right)^{\mathrm{T}} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ with $\mathbf{A} = \mathbf{H} \odot \mathbf{G}$. Ported to this framework, the following function is then used:

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, i_1, i_2, j_1, j_2) = e^{(1)}_{i_1 j_1} e^{(2)}_{i_2 j_2} + e^{(1)}_{i_2 j_2} e^{(2)}_{i_1 j_1} - e^{(1)}_{i_1 j_2} e^{(2)}_{i_2 j_1} - e^{(1)}_{i_2 j_1} e^{(2)}_{i_1 j_2}, \quad (25)$$

in which $\mathbf{E}^{(1)}$ and $\mathbf{E}^{(2)}$ are segmented versions of $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$, respectively, with dimensions $I \times J$. The matrices obtained by evaluating $\underline{z}$ in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ or $\{\mathbf{a}_1, \ldots, \mathbf{a}_K\} \times \{\mathbf{a}_1, \ldots, \mathbf{a}_K\}$ can then be stacked for all meaningful combinations of parameter values $1 \leq i_1 < i_2 \leq I$ and $1 \leq j_1 < j_2, \leq J$, yielding the tensors $\mathcal{Z}^{(x)}$ and $\mathcal{Z}^{(a)}$, respectively. It can be seen that the diagonals of the slices of $\mathcal{Z}^{(a)}$ are all zero, following from the fact that each column of $\mathbf{A}$ can be written as $\mathbf{a}_r = \mathbf{h}_r \odot \mathbf{g}_r$. Hence, the discussed procedure can be applied to recover $\mathbf{B}$ and subsequently $\mathbf{G}$ and $\mathbf{H}$.

Furthermore, in [69], a similar technique is used to obtain an algebraic algorithm for the multilinear rank-$(L_r, L_r, 1)$ decomposition. The ideas are generalized in [55] to allow the computation of a CPD of which none of the factor matrices have full column rank.

The separation of constant modulus signals as illustrated in Section VII-B1 has recently been extended to allow multi-modulus signals (such as 16-QAM signals), yielding a set of coupled CPDs [219]. Separation techniques based on the same approach as ACMA have also appeared for discrete / finite alphabet [220], constant power [221], binary [222] and on/off signals [223].

In Section VI, we have explored techniques to separate (evaluated) exponential functions and rational functions, allowing the recovery of $\mathbf{B}$. These models can be ported to this framework as well, recovering the separation matrix $\mathbf{W}$. For example, in the case of exponentials [161], $\mathcal{F}^{(x)}$ can then be constructed using the following function:

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, n) = 2u^{(1)}_n u^{(2)}_n - u^{(1)}_{n+1} u^{(2)}_{n-1} - u^{(1)}_{n-1} u^{(2)}_{n+1}$$

with $1 < n < N$. Again, it is easy to see that the slice diagonals of $\mathcal{F}^{(a)}$ are zero if and only if the vectors $\mathbf{a}_r$ are exponentials, as they satisfy the property $a^2_{nr} = a_{n-1,r} a_{n+1,r}$. Similar to the technique presented in Section VI-D, this technique is more flexible compared with the Hankelization and segmentation procedures and can be used if a number of samples are missing, if the signals are not equidistantly sampled, or for non-standard array shapes.

C. Direct construction of a tensor

Instead of concatenating a set of constructed matrices, one can directly construct a multiway tensor from a given matrix $\mathbf{X}$. Higher-order moments and cumulants, for example, are two well-known versions of higher-order statistics [224]. Although defined as higher-order derivatives of characteristic functions, they are most common in their explicit form as we will discuss. Moments are closely connected with N-grams from natural language processing [225]. We will illustrate the use of moments and N-grams for clustering in Section VII-C1 and make links with stacked outer products. Cumulants, on the other hand, are prevalent in independent component analysis as will be shown in Section VII-C2. The use of adjacency tensors in graph analysis is discussed in Section VII-C3, while neural networks have also clear links with tensors through score functions. The latter is covered in Section VII-C4.

Insert 18 — Unique matrix factorization: The approach for unique matrix factorization using the direct construction of a higher-order tensor typically consists of the following. Let the tensorization technique be defined by the function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$, and let $\mathcal{Z}^{(x)}$ and $\mathcal{Z}^{(a)}$ be the order-D tensorized versions of $\mathbf{X}$ and $\mathbf{A}$, respectively. Similar to the discussion in Insert 15, a multilinear function $\underline{z}$ allows one to write

$$\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \bullet_1 \mathbf{B} \cdots \bullet_D \mathbf{B}.$$

Under the assumption of diagonality of $\mathcal{Z}^{(a)}$ with non-zero diagonal entries, $\mathcal{Z}^{(x)}$ admits the rank-R CPD $\mathcal{Z}^{(x)} = [\![\mathbf{d}^{(a)}; \mathbf{B}, \ldots, \mathbf{B}]\!]$ with $d^{(a)}_r = z^{(a)}_{r \ldots r}$. This CPD is unique under mild conditions on $\mathbf{B}$ such that $\mathbf{B}$ (and subsequently $\mathbf{A}$) can be recovered.

1) Higher-order moments and N-grams: Higher-order statistics (HOS) have seen a broad use in statistical theory, signal processing and data analysis [203], [226], [227]. The original definition of HOS is based on higher-order derivatives of a characteristic function, while a more known version of the definition is based on the evaluation of a function $\underline{z}$ in a grid. We will show how these two versions are linked to each other. We will limit the discussion in this section to higher-order moments; higher-order cumulants are covered in Section VII-C2. The concepts apply to covariance matrices (Section VII-A2) as well, as covariance matrices are second-order versions of cumulants. Higher-order moments are also known as N-grams in natural language processing and computational linguistics as we will discuss [225].

We consider again the set of stochastic variables χ_k instead of the observations $\mathbf{x}_k$. The derivations are also valid for the latter case by substituting the expectancy operator with the mean. Let

$$\underline{g}(\upsilon) = \mathrm{E}\left\{e^{j\upsilon}\right\} \quad (26)$$

with υ a stochastic variable and let us substitute $\upsilon = \omega_1 \chi_1 + \ldots + \omega_K \chi_k$ such that the function

$$\underline{h}(\omega_1, \ldots, \omega_K) = \mathrm{E}\left\{e^{j(\omega_1 \chi_1 + \ldots + \omega_K \chi_K)}\right\} \quad (27)$$

is obtained. This function is also known as the first characteristic function. Order-D moments are now defined as the stacked order-D derivatives of $\underline{h}(\omega_1, \ldots, \omega_K)$ evaluated in $\omega_1, \ldots, \omega_K = 0$ and scaled by $1/j^D$. We have discussed this tensorization technique of stacking higher-order derivatives

in Section IV-C. Technically, we have the order-D moment $\mathcal{M} \in \mathbb{R}^{K \times \cdots \times K}$ with for all indices:

$$m_{k_1 \cdots k_D} = \frac{1}{j^D} \left. \frac{\partial^D \mathrm{E}\left\{e^{j(\omega_1 \chi_1 + \cdots + \omega_K \chi_K)}\right\}}{\partial \omega_{k_1} \cdots \partial \omega_{k_D}} \right|_{\omega_1, \ldots, \omega_D = 0} \tag{28}$$

Hence, each entry of $\mathcal{M}$ contains the order-D derivative of $\underline{h}$ to a combination of D variables from $\omega_1, \ldots, \omega_K$ in which repetition is allowed. This is the classical derivative-based definition of the higher-order moment. However, in practice, it can be cumbersome to compute these higher-order derivatives. Another version of the definition is much more established, which can be linked to the evaluation of a function $\underline{z}(\upsilon_1, \ldots, \upsilon_D)$ in a product grid.

Let us first compute the derivatives by applying the chain rule. Then, one can write

$$\mathcal{M} = \mathrm{E}\left\{ \left(\frac{1}{j^D} \left. \frac{\partial^D \underline{g}(\upsilon)}{\partial \upsilon^D} \right|_{\upsilon = 0} \right) (\boldsymbol{\chi} \otimes \cdots \otimes \boldsymbol{\chi}) \right\}.$$

The scalar term in the brackets is equal to 1. Hence, the following simple well-known explicit formula for the higher-order moment emerges with for all indices:

$$m_{k_1 \cdots k_D} = \mathrm{E}\left\{\chi_{k_1} \cdots \chi_{k_D}\right\}. \tag{29}$$

The attentive reader has now been able to detect two different types of tensorization in both definitions of $\mathcal{M}$. On the one hand, following the definition in (28), the tensor $\mathcal{M}$ consists of higher-order derivatives of the function $\underline{g}(\upsilon)$. On the other hand, following the definition in (29), $\mathcal{M}$ consists of evaluations of the function

$$\underline{z}(\upsilon^{(1)}, \ldots, \upsilon^{(D)}) = \mathrm{E}\left\{\upsilon^{(1)} \cdots \upsilon^{(D)}\right\} \tag{30}$$

in the grid $\{\chi_1, \ldots, \chi_K\} \times \cdots \times \{\chi_1, \ldots, \chi_K\}$. The two tensorization techniques are associated with each other because of the use of the linear combination substitution in (27). (30) can be seen as a simplified form of

$$\underline{z}(\upsilon^{(1)}, \ldots, \upsilon^{(D)}) = \left. \frac{1}{j^D} \frac{\partial^D \mathrm{E}\left\{e^{j(\omega_1 \upsilon^{(1)} + \cdots + \omega_D \upsilon^{(D)})}\right\}}{\partial \omega_1 \cdots \partial \omega_D} \right|_{\omega_d = 0}.$$

which is the polar form of $\underline{g}(\upsilon)$ obtained through polarization [105], [228]. The two different derivations leading to the same tensor $\mathcal{M}$ can then be summarized as follows:

1) Substitute υ in $\underline{g}(\upsilon)$ with $\omega_1 \chi_1 + \cdots + \omega_K \chi_K$ and compute the higher-order derivatives of $\underline{h}$ in zero for each combination of D variables ω_k or
2) Substitute υ in $\underline{g}(\upsilon)$ with $\omega_1 \upsilon^{(1)} + \cdots + \omega_D \upsilon^{(D)}$, compute the higher-order derivative to $\omega_1, \ldots, \omega_D$ in zero and evaluate $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ in the Cartesian product grid $\{\chi_1, \ldots, \chi_K\} \times \cdots \times \{\chi_1, \ldots, \chi_K\}$.

The same tensor emerges following both procedures.

In practice, often only observations of the stochastic variables are available. The previous expressions are still valid by substituting υ and χ with $\mathbf{u}$ and $\mathbf{x}$, respectively, and by replacing the expectancy operator by the mean. Procedures 1 and 2 in this discrete case are then visualized in Figure 14, respectively on the right and left.

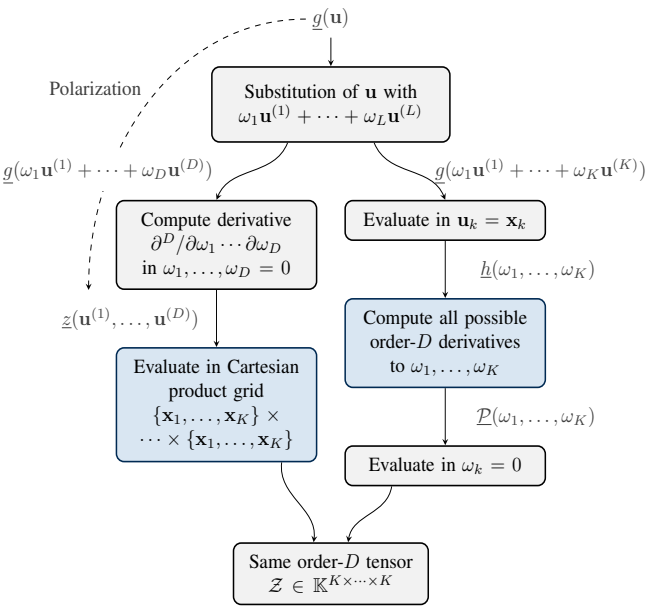


Fig. 14. Illustration of the order-D tensorization of K data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K \in \mathbb{K}^N$ based on a scalar function $\underline{g}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{K}^N$. Two different sequences of steps yield the same tensor $\mathcal{Z}$, each induced by two fundamentally different tensorization techniques (in blue). The functions $\underline{g}(\upsilon) = \mathrm{E}\left\{e^{j\upsilon}\right\}$ and $\underline{g}(\upsilon) = \log \mathrm{E}\left\{e^{j\upsilon}\right\}$ then lead to the higher-order moment $\mathcal{M}$ and higher-order cumulant $\mathcal{C}$, respectively, as discussed in Sections VII-C1 and VII-C2.

Although used in a different context, N-grams are mathematically equal to higher-order moments. N-grams are defined as the co-occurrence probabilities of objects, such as words in the same document or topic. By collecting the occurrence probabilities of words in each document or topic in a matrix $\mathbf{X}$, the N-grams are equal to the order-N moments of $\mathbf{X}$.

Insert 19 — Clustering: Let us recall the clustering problem from Insert 16. By computing, e.g., the third-order moment (or trigram) of $\mathbf{X}$, we obtain a CPD:

$$\mathcal{M}^{(x)} = \mathcal{M}^{(a)} \cdot_1 \mathbf{B} \cdot_2 \mathbf{B} \cdot_3 \mathbf{B} = \left[\!\left[\boldsymbol{\pi}^{(a)}; \mathbf{B}, \mathbf{B}, \mathbf{B} \right]\!\right], \tag{31}$$

with $\boldsymbol{\pi}^{(a)}$ the diagonal of $\mathcal{M}^{(a)}$. Indeed, if $\mathbf{A}$ is a selection matrix, $\mathcal{M}^{(a)}$ is diagonal. The vector $\boldsymbol{\pi}^{(a)}$ then contains the probabilities of each cluster (or topic in topic modeling) [14], [19], [20].

It can be seen that the higher-order moments are reduced versions of stacked higher-order outer products from Section VII-A1. A third-order moment, then, is obtained by applying a summation along the fourth mode of the tensor obtained by stacking third-order outer products. The CPD in (31) is then a direct consequence from the CPD obtained in (20).

Under the presence of noise, some additional measures can be taken. For example, the smallest eigenvalues of the covariance matrix of $\mathbf{X}$ give an indication of the noise variance [20]. By subtracting the noise contributions from the moments, the noise can be neutralized to some extent. It is not straightforward to incorporate the knowledge of

the noise statistics in stacked outer products from Section VII-A1.

2) Higher-order cumulants: In Section VII-C1, we have shown that higher-order moments can be defined as the higher-order derivatives of the first characteristic function, but can also be defined as the evaluation of some multivariate function in a grid. The same is true for higher-order cumulants, which are based on the function $\underline{g}(\upsilon) = \log \mathrm{E}\left\{e^{j\upsilon}\right\}$. Note the additional $\log$ compared to (26). After substitution with $\omega_1\upsilon_1 + \cdots + \omega_K\upsilon_K$, the second characteristic function is obtained:

$$\underline{h}(\omega_1, \ldots, \omega_K) = \log \mathrm{E}\left\{e^{j(\omega_1\chi_1 + \ldots + \omega_K\chi_K)}\right\}.$$

The order-D cumulant [229], [230] is defined as the stacking of all order-D derivatives of $\underline{h}(\omega_1, \ldots, \omega_K)$ evaluated in the origin. The second-order cumulant (also known as the covariance matrix) consists of the entries $c_{k_1k_2} = \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_2}\}$ with $\tilde{\chi}$ the zero-mean version of χ, while the entries of the fourth-order cumulant are defined as

$$\begin{aligned} c_{k_1\cdots k_4} = {} & \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_2}\tilde{\chi}_{k_3}\tilde{\chi}_{k_4}\} - \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_2}\}\mathrm{E}\{\tilde{\chi}_{k_3}\tilde{\chi}_{k_4}\} \\ & - \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_3}\}\mathrm{E}\{\tilde{\chi}_{k_2}\tilde{\chi}_{k_4}\} - \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_4}\}\mathrm{E}\{\tilde{\chi}_{k_2}\tilde{\chi}_{k_3}\}. \end{aligned}$$

Alternatively, $\mathcal{C}$ can be defined as the evaluation of the polarized form $\underline{z}$ of $\underline{g}(\upsilon)$ in the Cartesian product grid with

$$\begin{aligned} \underline{z}(\upsilon_1, \ldots, \upsilon_4) = {} & \mathrm{E}\{\tilde{\upsilon}_1\tilde{\upsilon}_2\tilde{\upsilon}_3\tilde{\upsilon}_4\} - \mathrm{E}\{\tilde{\upsilon}_1\tilde{\upsilon}_2\}\mathrm{E}\{\tilde{\upsilon}_3\tilde{\upsilon}_4\} \\ & - \mathrm{E}\{\tilde{\upsilon}_1\tilde{\upsilon}_3\}\mathrm{E}\{\tilde{\upsilon}_2\tilde{\upsilon}_4\} - \mathrm{E}\{\tilde{\upsilon}_1\tilde{\upsilon}_4\}\mathrm{E}\{\tilde{\upsilon}_2\tilde{\upsilon}_3\}. \quad (32) \end{aligned}$$

It is well-known that mutually independent stochastic variables $\chi_1, \ldots, \chi_K$ yield a diagonal cumulant [227], [231], but it less clear why. In contrast to moments, it can be seen that the diagonality emerges because of the presence of the $\log$ function. Indeed, by recalling that $\mathrm{E}\{\upsilon_1\upsilon_2\} = \mathrm{E}\{\upsilon_1\}\mathrm{E}\{\upsilon_2\}$ for mutually independent stochastic variables υ_1 and υ_2, $\underline{h}(\omega_1, \ldots, \omega_K)$ then boils down to a sum of univariate functions with

$$\underline{h}(\omega_1, \ldots, \omega_K) = \log \mathrm{E}\left\{e^{j\omega_1\chi_1}\right\} + \cdots + \log \mathrm{E}\left\{e^{j\omega_K\chi_K}\right\}.$$

We have seen in Section IV-C that a sum of univariate functions yields a diagonal higher-order derivative tensor.

Again, in the discrete case, the expectancy operator, υ and χ can be replaced by the mean[21], $\mathbf{u}$ and $\mathbf{x}$, respectively. The obtained sample cumulant $\hat{\mathcal{C}}$ will be approximately[22] diagonal for mutually independent signals.

Insert 20 — Independent component analysis: Because of the diagonality of cumulant tensors for mutually independent stochastic variables, cumulants are often used for the separation of non-Gaussian mutually independent signals in ICA [203]. This readily yields a CPD, as discussed in Insert 18. As the covariance matrix does not provide sufficient information, and as the third-order cumulant tensor is zero for symmetric distributions, fourth-order cumulants are most common. In early ICA algorithms, the covariance matrix was used in a PCA-based preprocessing step to obtain orthogonal factor matrices in the subsequent tensor decomposition [16], [233]. In the presence of (Gaussian) noise, however, it has been shown that the PCA step can introduce significant errors and prevents asymptotically unbiased estimates [234]. Furthermore, developments have led to well-established CPD algorithms that are able to cope well with non-orthogonal and possibly ill-conditioned factor matrices.

Rather than using only fourth-order cumulants [218], it can still be beneficial to combine these cumulants with covariance matrices (and other meaningful lower-order statistics) which typically suffer less from estimation errors [235], [236]. For example, the covariance matrix can be concatenated with the reshaped version of the fourth-order cumulant of size $K \times K \times K^2$, whereas coupled tensor decompositions might provide a more flexible approach. Second, it can be beneficial to assume the covariance matrix diagonal as unknown, as it consists of the largest noise contribution in the case of uncorrelated noise components. Note that a series of cumulants up to order D can also be stacked in a order-D tensor of dimensions $(K+1) \times \cdots \times (K+1)$ and subsequently decomposed as explained in Section IV-C.

In any case, sufficient importance should be placed on the weighting of the various statistics. A statistically optimal weighting exists, but can be hard to find. We refer the interested leader to [237], [238], while weighting techniques dealing with stacked covariance matrices from Section VII-A can be found in [239]–[241].

3) Graph adjacency tensors: Graphs are defined as sets of vertices or nodes and edges, the latter typically expressing relations between those nodes. Nodes can represent objects, persons, locations or events, amongst others. Graphs are often seen as abstractions of networks (such as social, computer or supply networks), and can be (un)weighted and/or (un)directed [242]. Although typically visualized graphically, cf. Figure 15, one can express a graph with K nodes uniquely with its square adjacency matrix $\mathbf{X} \in \mathbb{K}^{K\times K}$, which will be binary and symmetric for unweighted and undirected graphs, respectively. $\mathbf{X}$ contains the second-order structure of the graph, i.e., entry x_{ik} expresses the connection from node i to node k with $x_{ik} = 0$ denoting a non-existing connection. Hence, porting this concept to the framework in this section, a graph corresponds to a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$, with the kth vector denoting the connections to node k.

Tensors have appeared in graph theory before, typically in the form of stacked adjacency matrices of different graph instances. Each adjacency matrix then corresponds to the same set of nodes but to a different set of edges, typically forming a layered or time-evolving graph. For example, airports can be linked through different airlines [243], websites can be linked through a number of different keywords [244], or authors can

[21] Better estimators for the cumulant entries exist such as the k-statistics which are the minimum-variance unbiased estimators [232]

[22] Observe the difference with the diagonal property of higher-order moments. While a sample cumulant is only approximately diagonal for a finite number of samples of mutually independent variables, the higher-order moment is exactly diagonal for selection matrices.

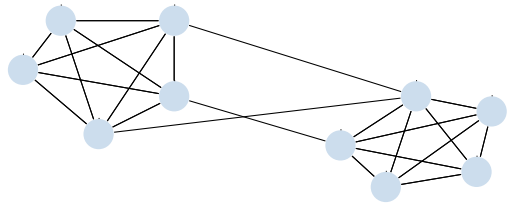


Fig. 15. An example of a graph with two clearly identifiable yet connected clusters.

be linked through different citations or collaborations [245]. In [209], [246], adjacency matrices corresponding to different time instances are stacked. The tensorization technique is then similar to the concept discussed in Section VII-A, consisting of the stacking of different (parameter-dependent) matrices.

On the other hand, tensors appear more naturally by considering higher-order graph structures, which can be entirely deduced from the adjacency matrix. $\mathbf{X}$ can then be tensorized to the adjacency tensor $\mathcal{Z} \in \mathbb{K}^{K\times K\times K}$, using suitable definitions. For example, for a binary symmetric $\mathbf{X}$, the triangle adjacency tensor consists of entries $z_{ijk} = 1$ if $x_{ij} = x_{ik} = x_{jk} = 1$ or if any two indices are equal and $z_{ijk} = 0$ otherwise, i.e., 'triangle' connections between nodes i, j and k are extracted [247], [248]. Other higher-order types such as stars correspond to other adjacency tensor definitions [247].

Adjacency tensors also form the basis of transition tensors corresponding to higher-order Markov chains. For example, the entries p_{ijk} of a transition tensor $\mathcal{P}$ then indicate the probability of transitioning to state i given the current state j and previous state k. Transition tensors have been used in applications such as multilinear pagerank [243], [249], [250].

Insert 21 — Graph clustering: A common objective in graph theory is the identification of specific clusters in graphs, i.e., well-connected subgraphs. Let us consider a graph with K nodes and R clusters, with the rth cluster containing K_r nodes. Nodes in the same cluster are assumed to be fully connected and there exist no connections between clusters. For $R = 2$, a so-called bipartite graph is obtained.

Let us construct the triangle tensor $\mathcal{Z}$. By ordering the nodes of the graph such that nodes of the same cluster are grouped, it can be seen that $\mathcal{Z}$ is a block diagonal tensor with each block consisting of all ones. Note that the tensor with all ones has rank 1 and can be written as $\mathbf{1}_K \otimes \mathbf{1}_K \otimes \mathbf{1}_K$. It is then easy to show that $\mathcal{Z}$ admits a rank-R CPD $\mathcal{Z} = [\![\mathbf{U}, \mathbf{U}, \mathbf{U}]\!]$ with factor matrix [247], [249], [251]

$$\mathbf{U} = \begin{bmatrix} \mathbf{1}_{I_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{I_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{I_R} \end{bmatrix}.$$

Even if the nodes are not reordered, the clusters can be easily extracted using a truncation procedure or data clustering techniques such as k-means [199]. In the non-ideal case where clusters are not fully connected and with connections between clusters, such as in Figure 15, the CPD holds only approximately, but still a suboptimal estiamte of $\mathbf{U}$ can be obtained [247]. The clusters can then easily be extracted.

4) Higher-order score functions: Higher-order score functions are similar to moments [252]. Given a stochastic vector $\boldsymbol{\chi} \in \mathbb{K}^K$ and its probability density function $\underline{p}(\mathbf{x})$ in function of an observation $\mathbf{x} \in \mathbb{K}^K$, the order-$D$ score function $\underline{\mathcal{S}}(\mathbf{x}) \in \mathbb{K}^{K\times\cdots\times K}$ is defined as

$$\underline{\mathcal{S}}(\mathbf{x}) = (-1)^D \frac{\underline{\mathcal{P}}(\mathbf{x})}{\underline{p}(\mathbf{x})}, \tag{33}$$

with $\underline{\mathcal{P}}$ consisting of the higher-order derivatives of $\underline{p}$ as defined in Section IV-C. Considering a particular model which maps an observation $\mathbf{x}$ of $\boldsymbol{\chi}$ to $\underline{y}(\mathbf{x})$, the so-called corresponding order-D cross-moment tensor $\mathcal{T} \in \mathbb{K}^{K\times\cdots\times K}$ is defined as

$$\mathcal{T} = \mathrm{E}\left\{\underline{y}(\mathbf{x})\underline{\mathcal{S}}(\mathbf{x})\right\}. \tag{34}$$

The order-D cross moment has special properties for specific types of models, e.g., for neural networks with a single hidden layer as we discuss in Insert 22

Insert 22 — Neural networks: Let us consider a given set of feature vectors with corresponding scalar labels. We will model the classifier using a neural network with a single hidden layer.

Note that the results can easily be generalized to vector labels. The neural network is modeled as follows:

$$\underline{y}(\mathbf{x}) = \mathbf{b}^{\mathrm{T}}\underline{\sigma}(\mathbf{A}^{\mathrm{T}}\mathbf{x}), \tag{35}$$

as depicted in Figure 16. The so-called activation function $\underline{\sigma}$ is evaluated element-wise and is assumed to be sufficiently differentiable. Note that constant terms can be added by considering an additional input $x_{K+1} = 1$, cf. the homogenization procedure from Section V-B2.

Let us assume N feature vectors are given which are the rows of the given matrix $\mathbf{X}$. The vector $\mathbf{x}$ in (33), (34) and (35) can then be substituted with the nth row of $\mathbf{X}$. Each feature vector can be seen as an observation of a stochastic vector $\boldsymbol{\chi}$, of which we know the probability density function $\underline{p}$. The nth feature vector corresponds with the label y_n. The goal consists of finding the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ given these observations. Typically, these are recovered by training the neural networks using methods such as back propagation [253].

Alternatively, let us construct the cross-tensor method $\mathcal{T}$. In practice, we have

$$\mathcal{T} = \frac{1}{N}\sum_n y_n \underline{S}((\mathbf{X}^{\mathrm{T}})_n).$$

Assuming that the labels y_n satisfy the model in (35), it has been proven that the order-D cross-moment tensor $\mathcal{T}$ from (34) admits a rank-R CPD of the simple form $\mathcal{T} = [\![\mathbf{A}, \ldots, \mathbf{A}]\!]$ [21], [252]. By constructing $\mathcal{T}$ and applying a CPD, the matrix $\mathbf{A}$ can readily be obtained.

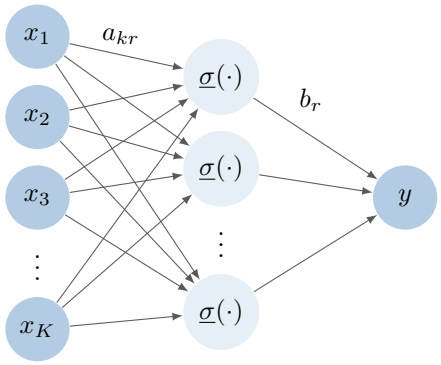


Fig. 16. Visualization of a neural network with one hidden layer. The input is an observation of the stochastic vector $\boldsymbol{\chi} \in \mathbb{K}^K$. The neural network models a scalar label y in function of the input, but this can be extended easily to a vector output.

Hence, computing the CPD is equivalent to training the neural network. Note that algebraic CPD algorithms exist which do not suffer from local optima, unlike optimization-based algorithms. Even in the case that the labels do not perfectly follow the model, algebraic CPD algorithms can still provide suboptimal estimates of $\mathbf{A}$ and can good provide good initializations for optimization-based techniques.

The appearance of the CPD can be understood from the fact that the function $\underline{\sigma}$ is applied element-wise, as is also the case in the decoupling model of multivariate functions from Insert 3. Indeed, it can be shown that the neural network model is a specific instance of the model in (4). However, the technique from Insert 3 cannot be applied in the case of neural networks as the higher-order derivatives of $\underline{y}$ are typically unknown. The method based on score functions considers the derivatives of the probability density function of the inputs rather than the derivatives of $\underline{y}$.

5) Other tensorization techniques and further notes on polarization: Higher-order statistics, adjacency tensors and score functions might be the most well-known techniques that map a given matrix $\mathbf{X}$ to a natural multiway tensor. As discussed, each function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ has the availability of tensorizing a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$ to an order-D tensor. To be able to apply tensor tools in a subsequent step, the function $\underline{z}$ typically needs to be multilinear.

The polarization-based discussion in Section VII-C1 provides a novel framework as it shows that each function $\underline{g}(\mathbf{u})$ leads to a function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$, called the polar form of $\underline{g}(\mathbf{u})$, which can map a given set of vectors to a tensor. Furthermore, it is well known in algebra that the polar form is multilinear [105], [228]. We reproduce a brief proof of this in Appendix A. This is not unimportant: it shows that a meaningful multilinear tensorization technique can be obtained by choosing a suitable function $\underline{g}(\mathbf{u})$ or, considering stochastic variables, $\underline{g}(\upsilon)$.

We have discussed that both higher-order moments and higher-order cumulants originate from a particular choice of $\underline{g}(\upsilon)$, namely from $\underline{g}(\upsilon) = \mathrm{E}\left\{e^{j\upsilon}\right\}$ and $\underline{g}(\upsilon) = \log \mathrm{E}\left\{e^{j\upsilon}\right\}$, respectively. Although not presented in such a way, the determinant function $\underline{g}(\mathbf{U}) = \det(\mathbf{U})$ lies at the core of the algebraic algorithm developed in [55] to compute a CPD in which none of the factor matrices have full column rank. This function gives rise to the mixed discriminant function $\underline{z}(\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)})$ with

$$\underline{z}(\cdot) = \sum_{d=1}^{D} \sum_{\substack{1 \leq i_d \leq D \\ i_1 < \cdots < i_D}} (-1)^{(D-d)} \det(\mathbf{U}^{(i_1)} + \cdots + \mathbf{U}^{(i_d)}). \tag{36}$$

More correctly, the vector function $\underline{\mathbf{g}}(\mathbf{U}) = \mathrm{vec}\left(\mathbf{C}_M(\mathbf{U})\right)$ is applied which leads to the (vectorized) polarized compound matrix function $\underline{\boldsymbol{z}}(\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)})$. The latter satisfies a similar expression as in (36) but with $\det(\cdot)$ replaced by $\mathrm{vec}\left(\mathbf{C}_D(\cdot)\right)$. It can be seen that, for $D = 2$, $\underline{\boldsymbol{z}}$ contains the scalar functions defined in (25). Furthermore, it should not come as a surprise that (36) and (32) (the definition of the fourth-order cumulant) have a very similar structure, as they are both derived in a similar way from a function $\underline{g}$.

VIII. Further discussions and future work

A number of different concepts have been proposed on how a vector or matrix can be mapped to a tensor. A tensorization technique should not uniquely be identified with a single concept, as the concepts can also be joined. For example, spatio-temporal cumulants merge the concepts of parameter variation from Section VII-A and the direct construction of a cumulant from Section VII-C2. Hence, the spatio-temporal cumulants not only take into account the non-Gaussian character of the signals but also their non-stationarity. For example, the spatio-temporal generalization of the fourth-order cumulant yields a tensor of order seven, each entry depending on a combination of four signals and three time lags. Spatio-temporal cumulants in combination with tensor decompositions are particularly useful for blind system identification [254]–[256].

Furthermore, tensorization techniques can also be combined and applied simultaneously. Given a matrix $\mathbf{X}$, twofold tensorization applies a tensorization technique on each mode of $\mathbf{X}$ simultaneously. This can be used to incorporate assumptions on both $\mathbf{A}$ and $\mathbf{B}$ given the model $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$. This twofold tensorization then results in a tensor of order four or higher. For example, in the context of instantaneous BSS, twofold segmentation can be used to exploit the Kronecker structure of both $\mathbf{A}$ (the source level) and $\mathbf{B}$ (the mixing level) [26]. Similarly, if $\mathbf{B}$ has a Kronecker structure and $\mathbf{A}$ contains mutually independent and non-Gaussian signals, a combination of segmentation and higher-order cumulants can be used in a context of ICA [257]. Given a matrix $\mathbf{X}$, the construction of a compound matrix as discussed in Insert 2 can also be seen as twofold tensorization.

Many of the tensorization techniques have been illustrated based on a number of applications such as ICA, clustering and the training of neural networks. While relatively simple underlying models have been considered, advanced models might be more appropriate in a real-life setting. Rather than instantaneous ICA, single-membership cluster models or single-layer neural networks, one can consider convolutional ICA, mixed-membership cluster models, or neural networks with multiple

hidden layers (deep neural networks) or with feedback (recurrent neural networks). While tensorization techniques have already seen a broad use in the convolutive generalization of ICA [254]–[256], one can imagine that the power of tensor tools can also be employed in the context of the other generalizations. A limited number of results have already appeared, discussing the use of CPD in mixed-membership models [251] and the hierarchical Tucker decomposition for deep neural networks [22].

IX. Conclusion

This paper has covered a tensorization framework and has surveyed a large number of tensorization techniques which have been presented rather disparately in the literature. Tensorization is defined as the mapping of a vector or matrix to a tensor or, more generally, of mapping lower-order data to higher-order data. It has been illustrated how the techniques are fundamental to tensor-based methods for deterministic blind signal separation, independent component analysis, data clustering and topic modeling, graph clustering and the training of neural networks, among others. Links between different tensorization techniques have been established, such as between segmentation and time–frequency and time–scale representations, between higher-order statistics and mixed discriminants through a process called polarization, and between the analytical constant modulus algorithm and other algebraic variety-based methods. This polarization procedure allows the development of new multilinear tensorization techniques. Furthermore, we have recognized that a number of problems that are ubiquitous in engineering can be well represented by tensors. For example, the approximation of a non-linear function using a power series expansion is equivalent to representing the non-linear function by a tensor, and finding the global optimum of a polynomial within a finite interval is equivalent to finding the best symmetric rank-1 approximation of a tensor.

Acknowledgment

The authors would like to thank Ignat Domanov for the insights on algebraic geometry and polarization, and Marc Van Barel for the discussions about Löwner tensors.

Appendix

Theorem 1. *Each possibly non-linear function $\underline{g}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{K}^N$ of which the order-D derivatives exist in $\mathbf{u} = \mathbf{0}$ leads to a multilinear mapping $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ of the form*

$$\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) = \left. \frac{\partial^D \underline{g}(\omega_1 \mathbf{u}^{(1)} + \cdots + \omega_D \mathbf{u}^{(D)})}{\partial \omega_1 \cdots \partial \omega_D} \right|_{\omega_d = 0}.$$

Proof. By applying the chain rule, one can see that

$$\left. \frac{\partial \underline{g}(\omega_1 \mathbf{u}^{(1)} + \cdots + \omega_D \mathbf{u}^{(D)})}{\partial \omega_1} \right|_{\mathbf{0}} = \mathbf{u}^{(1)^{\mathrm{T}}} \nabla \underline{g}|_{\mathbf{0}}$$

with $\nabla \underline{g}$ the gradient of $\underline{g}(\mathbf{u})$. Recursively applying the other $D-1$ derivatives, it follows that

$$\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) = \mathcal{P} \bullet_1 \mathbf{u}^{(1)^{\mathrm{T}}} \bullet_2 \mathbf{u}^{(2)^{\mathrm{T}}} \cdots \bullet_D \mathbf{u}^{(D)^{\mathrm{T}}}, \quad (37)$$

with $\mathcal{P}$ the order-D symmetric tensor of size $N \times \cdots \times N$ consisting of the higher-order derivatives of $\underline{g}(\mathbf{u})$ in $\mathbf{u} = \mathbf{0}$ as defined in (3). The function $\underline{z}$ from (37) is of the form in (6) and is therefore a multilinear mapping. □

In algebra, typically, a polynomial function $\underline{g}(\mathbf{u})$ is considered. However, $\underline{g}(\mathbf{u})$ can be non-linear as well, as long as the order-D derivatives are defined and can be computed in the origin.

Corollary 1. *Each function $\underline{g}(\mathbf{u})$ forms the basis of a multilinear order-D tensorization technique. Given vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$, an order-$D$ tensor of size $K \times \cdots \times K$ can be obtained by evaluating the corresponding polar form $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \cdots \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$.*

References

[1] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Studies in Applied Mathematics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[2] ——, "Multiple invariants and generalized rank of a p-way matrix or tensor," *Studies in Applied Mathematics*, vol. 7, no. 1-4, pp. 39–79, 1928.

[3] R. B. Cattell, "Parallel proportional profiles and other principles for determining the choice of factors by rotation," *Psychometrika*, vol. 9, no. 4, pp. 267–283, 1944.

[4] ——, "The three basic factor-analytic research designs — Their interrelations and derivatives." *Psychological bulletin*, vol. 49, no. 5, p. 499, 1952.

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

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

[7] R. A. Harshman, "Determination and proof of minimum uniqueness conditions for PARAFAC1," *UCLA Working Papers in Phonetics*, vol. 22, no. 111-117, p. 3, 1972.

[8] J. D. 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, 1970.

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

[10] A. K. Smilde, R. Bro, P. Geladi, and J. Wiley, *Multi-way analysis with applications in the chemical sciences*. Wiley Chichester, UK, 2004.

[11] P. Kroonenberg, *Applied multiway data analysis*. Wiley-Interscience, 2008, vol. 702.

[12] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[13] A. Cichocki, D. Mandic, A. H. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[14] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, July 2017.

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

[16] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proceedings F — Radar and Signal Processing*, vol. 140, no. 6, pp. 362–370, 1993.

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

[18] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[19] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2773–2832, 2014.

[20] D. Hsu and S. M. Kakade, "Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions," in *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. ACM, 2013, pp. 11–20.

[21] M. Janzamin, H. Sedghi, and A. Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv preprint arXiv:1506.08473*, 2016.

[22] N. Cohen, O. Sharir, and A. Shashua, "On the expressive power of deep learning: A tensor analysis," in *Proceedings of the 29th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49, June 2016, pp. 698–728.

[23] L. Cohen, "Time–frequency distributions — A review," *Proceedings of the IEEE*, vol. 77, no. 7, pp. 941–981, 1989.

[24] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[25] I. Daubechies, "The wavelet transform, time–frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[26] M. Boussé, O. Debals, and L. De Lathauwer, "A tensor-based method for large-scale blind source separation using segmentation," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 346–358, Jan 2017.

[27] B. Khoromskij, "Tensors-structured numerical methods in scientific computing: Survey on recent advances," *Chemometrics and Intelligent Laboratory Systems*, vol. 110, no. 1, pp. 1–19, 2012.

[28] G. H. Golub and C. F. Van Loan, *Matrix computations*, 4th ed. Johns Hopkins University Press, 2013.

[29] E. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[30] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[31] L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven, "An algorithm for generic and low-rank specific identifiability of complex tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 4, pp. 1265–1287, 2014.

[32] M. Sørensen and L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part I: Uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 496–522, 2015.

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

[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," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 855–875, 2013.

[35] ——, "On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part II: Uniqueness of the overall decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 876–903, 2013.

[36] ——, "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm," *Linear Algebra and its Applications*, vol. 513, pp. 342–375, 2017.

[37] ——, "Generic uniqueness conditions for the canonical polyadic decomposition and INDSCAL," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 4, pp. 1567–1589, 2015.

[38] L. Chiantini and G. Ottaviani, "On generic identifiability of 3-tensors of small rank," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 3, pp. 1018–1037, 2012.

[39] A. Stegeman, "On uniqueness of the nth order tensor decomposition into rank-1 terms with linear independence in one mode," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2498–2516, 2010.

[40] M. Haardt and J. A. Nossek, "Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems," *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 161–169, 1998.

[41] 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 Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

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

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

[44] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms — Part I: Lemmas for partitioned matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008.

[45] ——, "Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.

[46] L. De Lathauwer and D. Nion, "Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.

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

[48] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[49] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Annals of Physics*, vol. 349, pp. 117–158, 2014.

[50] L. Grasedyck, "Polynomial approximation in hierarchical Tucker format by vector tensorization," Apr. 2010.

[51] W. Hackbusch, *Tensor spaces and numerical tensor calculus*, ser. Springer Series in Computational Mathematics. Springer Science & Business Media, 2012, vol. 42.

[52] S. E. Leurgans, R. T. Ross, and R. B. Abel, "A decomposition for three-way arrays," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 4, pp. 1064–1083, 1993.

[53] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *Journal of Chemometrics*, vol. 4, no. 1, pp. 29–45, 1990.

[54] L. De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 642–666, 2006.

[55] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 2, pp. 636–660, 2014.

[56] A.-H. Phan, P. Tichavský, and A. Cichocki, "Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4834–4846, 2013.

[57] E. Acar, D. Dunlavy, and T. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *Journal of Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.

[58] 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 Journal on Optimization*, vol. 23, no. 2, pp. 695–720, 2013.

[59] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 284–295, 2016.

[60] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. (2016, March) Tensorlab 3.0. [Online]. Available: http://www.tensorlab.net

[61] C. Andersson and R. Bro, "The N-way Toolbox for MATLAB," *Chemometrics and Intelligent Laboratory Systems*, vol. 52, no. 1, pp. 1 – 4, 2000.

[62] B. W. Bader, T. G. Kolda *et al.* (2012, January) MATLAB Tensor Toolbox version 2.5. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

[63] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured data fusion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 586–600, June 2015.

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

[65] D. Lahat, T. Adalı, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, Sept 2015.

[66] E. Acar, R. Bro, and A. Smilde, "Data fusion in metabolomics using coupled matrix and tensor factorizations," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1602–1620, Sept 2015.

[67] K. Y. Yılmaz, A. T. Cemgil, and U. Simsekli, "Generalised coupled tensor factorisation," in *Advances in Neural Information Processing Systems (NIPS)*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011, pp. 2151–2159.

[68] M. Sørensen and L. De Lathauwer, "Multidimensional harmonic retrieval via coupled canonical polyadic decomposition — Part I: Model and identifiability," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 517–527, Jan 2017.

[69] M. Sørensen, I. Domanov, and L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part II: Algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 3, pp. 1015–1045, 2015.

[70] Y. Yu, J. Jin, F. Liu, and S. Crozier, "Multidimensional compressed sensing MRI using tensor decomposition-based sparsifying transform," *PloS one*, vol. 9, no. 6, pp. 1–13, 2014.

[71] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 71–79, September 2014.

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

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

[74] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[75] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5423–5436, 2015.

[76] C.-I. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Transactions on Information Theory*, vol. 46, no. 5, pp. 1927–1932, Aug 2000.

[77] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 67–81, 2014.

[78] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[79] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[80] J. Bennett, S. Lanning *et al.*, "The Netflix prize," in *Proceedings of KDD cup and workshop*. New York, NY, USA, 2007, pp. 3–6.

[81] I. Ovejero-Lopez, R. Bro, and W. Bredie, "Univariate and multivariate modelling of flavour release in chewing gum using time-intensity: A comparison of data analytical methods," *Food Quality and Preference*, vol. 16, no. 4, pp. 327–343, 2005.

[82] R. Bro and H. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *Journal of Chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.

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

[84] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of EEG signals: A brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015.

[85] E. Karahan, P. A. Rojas-Lopez, M. L. Bringas-Vega, P. A. Valdes-Hernandez, and P. A. Valdes-Sosa, "Tensor analysis and fusion of multimodal brain images," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1531–1559, 2015.

[86] R. Bro, "PARAFAC. Tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[87] C. M. Andersen and R. Bro, "Practical aspects of PARAFAC modeling of fluorescence excitation–emission data," *Journal of Chemometrics*, vol. 17, no. 4, pp. 200–215, 2003.

[88] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen, "Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone," *NeuroImage*, vol. 37, no. 3, pp. 844–854, 2007.

[89] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition (CVPR)*, 1991, pp. 586–591.

[90] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear (tensor) ICA and dimensionality reduction," in *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, ser. Lecture Notes in Computer Science, vol. 4666. Springer Berlin / Heidelberg, 2007, pp. 818–826.

[91] ——, "Multilinear analysis of image ensembles: Tensorfaces," in *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., vol. 2350. Springer Berlin / Heidelberg, 2002, pp. 447–460.

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

[93] T. Jiang and N. Sidiropoulos, "A direct blind receiver for SIMO and MIMO OFDM systems subject to unknown frequency offset and multipath," in *Proceedings of the 4th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2003, pp. 358–362.

[94] G. Beylkin and M. Mohlenkamp, "Algorithms for numerical analysis in high dimensions," *SIAM Journal on Scientific Computing*, vol. 26, no. 6, pp. 2133–2159, 2005.

[95] G. Beylkin, J. Garcke, and M. J. Mohlenkamp, "Multivariate regression and machine learning with sums of separable functions," *SIAM Journal on Scientific Computing*, vol. 31, no. 3, pp. 1840–1857, 2009.

[96] M. J. Mohlenkamp and L. Monzón, "Trigonometric identities and sums of separable functions," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 65–69, 2005.

[97] P. Dreesen, M. Schoukens, K. Tiels, and J. Schoukens, "Decoupling static nonlinearities in a parallel Wiener–Hammerstein system: A first-order approach," in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2015, pp. 987–992.

[98] F. Giri and E.-W. Bai, *Block-oriented nonlinear system identification*. Springer, 2010, vol. 1.

[99] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon, "Identification of nonlinear systems using polynomial nonlinear state space models," *Automatica*, vol. 46, no. 4, pp. 647–656, 2010.

[100] P. Dreesen, M. Ishteva, and J. Schoukens, "Decoupling multivariate polynomials using first-order information and tensor decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 864–879, 2015.

[101] V. Strassen, "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, no. 4, pp. 354–356, August 1969.

[102] J. Landsberg, "The border rank of the multiplication of 2×2 matrices is seven," *Journal of the American Mathematical Society*, vol. 19, no. 2, pp. 447–459, 2006.

[103] D. Hilbert and B. Sturmfels, *Theory of algebraic invariants*. Cambridge University Press, 1993.

[104] A. Cayley, "XXVIII — On the theory of the analytical forms called trees," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 13, no. 85, pp. 172–176, 1857.

[105] J. Landsberg, *Tensors: Geometry and applications*. American Mathematical Society Providence, RI, USA, 2012, vol. 128.

[106] L. Oeding and G. Ottaviani, "Eigenvectors of tensors and algorithms for Waring decomposition," *Journal of Symbolic Computation*, vol. 54, pp. 9–35, 2013.

[107] L. Mordell, "A new Waring's problem with squares of linear forms," *The Quarterly Journal of Mathematics*, vol. 1, pp. 276–288, 1930.

[108] F. Ayres, L. G. Díez, and Á. G. Vázquez, *Matrices*. McGraw–Hill New York, 1962.

[109] P. Comon and B. Mourrain, "Decomposition of quantics in sums of powers of linear forms," *Signal Processing*, vol. 53, no. 2, pp. 93–107, 1996.

[110] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, "Symmetric tensors and symmetric tensor rank," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1254–1279, 2008.

[111] S. Rendle, "Factorization machines," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, 2010, pp. 995–1000.

[112] L. Qi, "Eigenvalues of a real supersymmetric tensor," *Journal of Symbolic Computation*, vol. 40, no. 6, pp. 1302 – 1324, 2005.

[113] L. Qi, F. Wang, and Y. Wang, "Z-eigenvalue methods for a global polynomial optimization problem," *Mathematical Programming*, vol. 118, no. 2, pp. 301–316, 2009.

[114] E. Kofidis and P. A. Regalia, "On the best rank-1 approximation of higher-order supersymmetric tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 3, pp. 863–884, 2002.

[115] L. Qi and Z. Luo, *Tensor analysis: Spectral theory and special tensors*. SIAM, 2017.

[116] B. Sturmfels, *Solving systems of polynomial equations*, ser. CBMS Regional Conference Series in Mathematics. American Mathematical Society, 2002, vol. 97.

[117] B. Buchberger, "Gröbner bases and applications," in *London Mathematical Society Lecture Note Series*, B. Buchberger and F. Winkler, Eds., vol. 251. Cambridge, 1998.

[118] K. Hägglöf, P. O. Lindberg, and L. Svensson, "Computing global minima to polynomial optimization problems using Gröbner bases," *Journal of Global Optimization*, vol. 7, no. 2, pp. 115–125, 1995.

[119] K. Batselier, P. Dreesen, and B. D. Moor, "On the null spaces of the Macaulay matrix," *Linear Algebra and its Applications*, vol. 460, pp. 259 – 289, 2014.

[120] L. Sorber, M. Van Barel, and L. De Lathauwer, "Numerical solution of bivariate and polyanalytic polynomial systems," *SIAM Journal on Numerical Analysis*, vol. 52, no. 4, pp. 1551–1572, 2014.

[121] A.-H. Phan, P. Tichavský, and A. Cichocki, "Blind source separation of single channel mixture using tensorization and tensor diagonalization," in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science, vol. 10169. Springer Berlin / Heidelberg, 2017, pp. 36–46.

[122] O. Debals and L. De Lathauwer, "Stochastic and deterministic tensorization for blind signal separation," in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science, vol. 9237. Springer Berlin / Heidelberg, August 2015, pp. 3–13.

[123] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[124] T. Jiang, N. Sidiropoulos, and J. M. ten Berge, "Almost-sure identifiability of multidimensional harmonic retrieval," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1849–1859, 2001.

[125] S.-Y. Kung, K. S. Arun, and D. B. Rao, "State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem," *Journal of the Optical Society of America*, vol. 73, no. 12, pp. 1799–1811, 1983.

[126] D. Vandevoorde, "A fast exponential decomposition algorithm and its applications to structured matrices," Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, NY, 1998.

[127] J.-M. Papy, L. De Lathauwer, and S. Van Huffel, "Exponential data fitting using multilinear algebra: The single-channel and multi-channel case," *Numerical Linear Algebra with Applications*, vol. 12, no. 8, pp. 809–826, 2005.

[128] Y. Song and L. Qi, "Infinite and finite dimensional Hilbert tensors," *Linear Algebra and Its Applications*, vol. 451, pp. 1–14, 2014.

[129] W. Ding, L. Qi, and Y. Wei, "Fast Hankel tensor–vector product and its application to exponential data fitting," *Numerical Linear Algebra with Applications*, vol. 22, no. 5, pp. 814–832, 2015.

[130] S. Feldmann and G. Heinig, "Vandermonde factorization and canonical representations of block Hankel matrices," *Linear Algebra and its Applications*, vol. 241, pp. 247–278, 1996.

[131] D. W. Tufts and R. Kumaresan, "Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 975–989, 1982.

[132] M. Osborne, "Some special nonlinear least squares problems," *SIAM Journal on Numerical Analysis*, vol. 12, no. 4, pp. 571–592, 1975.

[133] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 814–824, 1990.

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

[135] J.-M. Papy, "Subspace-based exponential data fitting using linear and multilinear data," Ph.D. dissertation, KU Leuven, 2005.

[136] N. Vervliet, O. Debals, and L. De Lathauwer, "Tensorlab 3.0 — Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization," in *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, Nov 2016, pp. 1733–1738.

[137] ——, "Exploiting efficient data representations in large-scale tensor decompositions," 2017, Technical Report 16–174, ESAT–STADIUS, KU Leuven, Belgium.

[138] R. Badeau and R. Boyer, "Fast multilinear singular value decomposition for structured tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1008–1021, 2008.

[139] Q. Zhang, M. Berry, B. Lamb, and T. Samuel, "A parallel nonnegative tensor factorization algorithm for mining global climate data," in *Proceedings of the International Conference on Computational Science (ICCS)*. Springer, 2009, pp. 405–415.

[140] M. Boussé, O. Debals, and L. De Lathauwer, "Tensor-based large-scale blind system identification using segmentation," September 2016, Technical Report 16–109, ESAT–STADIUS, KU Leuven, Belgium.

[141] M. Alier, M. Felipe-Sotelo, I. Hernández, and R. Tauler, "Variation patterns of nitric oxide in Catalonia during the period from 2001 to 2006 using multivariate data analysis methods," *Analytica Chimica Acta*, vol. 642, no. 1, pp. 77–88, 2009.

[142] M. Alier, M. Felipe, I. Hernández, and R. Tauler, "Trilinearity and component interaction constraints in the multivariate curve resolution investigation of NO and O_3 pollution in Barcelona," *Analytical and Bioanalytical Chemistry*, vol. 399, no. 6, pp. 2015–2029, 2011.

[143] B. N. Khoromskij, "$\mathcal{O}(d \log N)$-quantics approximation of n-d tensors in high-dimensional numerical modeling," *Constructive Approximation*, vol. 34, no. 2, pp. 257–280, 2011.

[144] I. V. Oseledets, "Approximation of $2^d \times 2^d$ matrices using tensor decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2130–2145, 2010.

[145] G. Goovaerts, O. De Wel, B. Vandenberk, R. Willems, and S. Van Huffel, "Detection of irregular heartbeats using tensors," in *Proceedings of the Computing in Cardiology Conference (CinC)*. IEEE, 2015, pp. 573–576.

[146] G. Goovaerts, B. Vandenberk, R. Willems, and S. V. Huffel, "Tensor-based detection of T Wave Alternans using ECG," in *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 6991–6994.

[147] M. Boussé, G. Goovaerts, N. Vervliet, O. Debals, S. Van Huffel, and L. De Lathauwer, "Irregular heartbeat classification using Kronecker product equations," in *Proceedings of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Jul 2017, to appear.

[148] K. Löwner, "Über monotone Matrixfunktionen," *Mathematische Zeitschrift*, vol. 38, no. 1, pp. 177–216, 1934.

[149] A. C. Antoulas and D. C. Sorensen, "Approximation of large-scale dynamical systems: An overview," *International Journal of Applied Mathematics and Computer Science*, vol. 11, pp. 1093–1121, 2001.

[150] A. C. Antoulas, A. C. Ionita, and S. Lefteriu, "On two-variable rational interpolation," *Linear Algebra and its Applications*, vol. 436, no. 8, pp. 2889–2915, 2012.

[151] A. C. Antoulas and B. D. O. Anderson, "On the scalar rational interpolation problem," *IMA Journal of Mathematical Control and Information*, vol. 3, no. 2-3, pp. 61–88, 1986.

[152] V. Belevitch, "Interpolation matrices," *Philips Research Reports*, vol. 25, pp. 337–369, 1970.

[153] M. Fiedler, "Hankel and Löwner matrices," *Linear Algebra and its Applications*, vol. 58, pp. 75–95, 1984.

[154] Z. Vavřín, "A unified approach to Löwner and Hankel matrices," *Linear Algebra and its Applications*, vol. 143, pp. 171–222, 1991.

[155] ——, "Confluent Cauchy and Cauchy-Vandermonde matrices," *Linear Algebra and its Applications*, vol. 258, pp. 271–293, 1997.

[156] O. Debals, M. Van Barel, and L. De Lathauwer, "Löwner-based blind signal separation of rational functions with applications," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 1909–1918, April 2016.

[157] O. Debals, L. De Lathauwer, and M. Van Barel, "About higher-order Löwner tensors," April 2017, Technical Report 17–98, ESAT–STADIUS, KU Leuven, Belgium.

[158] L. De Lathauwer, "Block component analysis, a new concept for blind source separation," in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, vol. 7191, pp. 1–8.

[159] B. Hunyadi, D. Camps, L. Sorber, W. Van Paesschen, M. De Vos, S. Van Huffel, and L. De Lathauwer, "Block term decomposition for modelling epileptic seizures," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–19, 2014.

[160] M. Boussé, O. Debals, and L. De Lathauwer, "A tensor-based method for large-scale blind system identification using segmentation," in *Proceedings of the 24th IEEE European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 2015–2019.

[161] M. Sørensen and L. De Lathauwer, "Coupled tensor decompositions and monomial structure in array processing," 2015, Technical Report 15–152, ESAT–STADIUS, KU Leuven, Belgium.

[162] Z. Liu, Y. Xin, and G. B. Giannakis, "Space–time–frequency-coded OFDM over frequency-selective fading channels," *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2465–2476, 2002.

[163] H. Becker, P. Comon, L. Albera, M. Haardt, and I. Merlet, "Multi-way space–time–wave-vector analysis for EEG source separation," *Signal Processing*, vol. 92, no. 4, pp. 1021 – 1031, 2012.

[164] H. Becker, L. Albera, P. Comon, R. Gribonval, F. Wendling, and I. Merlet, "Brain-source imaging: From sparse to tensor models," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 100–112, Nov 2015.

[165] E. Acar, C. Aykut–Bingol, H. Bingol, R. Bro, and B. Yener, "Multiway analysis of epilepsy tensors," *Bioinformatics*, vol. 23, no. 13, pp. i10–i18, 2007.

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

[167] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[168] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time–frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[169] S. Rickard, R. Balan, and J. Rosca, "Real-time time–frequency based blind source separation," in *Proceedings of the third International Conference on Independent Component Analysis and Signal Separation (ICA)*, 2001.

[170] F. Abrard and Y. Deville, "A time–frequency blind signal separation method applicable to underdetermined mixtures of dependent sources," *Signal Processing*, vol. 85, no. 7, pp. 1389–1403, 2005.

[171] M. Zibulevsky and B. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural computation*, vol. 13, no. 4, pp. 863–882, 2001.

[172] J. B. Allen and L. R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[173] W. Koenig, H. Dunn, and L. Lacy, "The sound spectrograph," *The Journal of the Acoustical Society of America*, vol. 18, no. 1, pp. 19–49, 1946.

[174] F. Miwakeichi, E. Martınez-Montes, P. A. Valdés-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.

[175] M. Mørup, L. K. Hansen, and S. M. Arnfred, "ERPWAVELAB: A toolbox for multi-channel analysis of time–frequency transformed event related potentials," *Journal of Neuroscience Methods*, vol. 161, no. 2, pp. 361–368, 2007.

[176] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[177] A. L. F. de Almeida, G. Favier, and L. R. Ximenes, "Space–time–frequency (STF) MIMO communication systems with blind receiver based on a generalized PARATUCK2 model," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 1895–1909, 2013.

[178] E. Wigner, "On the quantum correction for thermodynamic equilibrium," *Physical review*, vol. 40, no. 5, p. 749, 1932.

[179] J. Ville *et al.*, "Théorie et applications de la notion de signal analytique," *Cables et transmission*, vol. 2, no. 1, pp. 61–74, 1948.

[180] H.-I. Choi and W. J. Williams, "Improved time–frequency representation of multicomponent signals using exponential kernels," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 6, pp. 862–871, 1989.

[181] J. G. Kirkwood, "Quantum statistics of almost classical assemblies," *Physical Review*, vol. 44, no. 1, p. 31, 1933.

[182] A. Rihaczek, "Signal energy distribution in time and frequency," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 369–374, 1968.

[183] I. Daubechies *et al.*, *Ten lectures on wavelets*. SIAM, 1992, vol. 61.

[184] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2207–2232, 1992.

[185] C. Guerrero, A. M. Trigueros, and J. I. Franco, "Time–frequency EEG analysis in epilepsy: What is more suitable?" in *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology*, 2005, pp. 202–207.

[186] M. Weis, F. Römer, M. Haardt, D. Jannek, and P. Husar, "Multi-dimensional space–time–frequency component analysis of event related EEG data using closed-form PARAFAC," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 349–352.

[187] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, "Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG," *NeuroImage*, vol. 29, no. 3, pp. 938–947, 2006.

[188] M. Latka, Z. Was, A. Kozik, and B. J. West, "Wavelet analysis of epileptic spikes," *Physical Review E*, vol. 67, no. 5, p. 052902, 2003.

[189] W. Deburchgraeve, P. Cherian, M. De Vos, R. Swarte, J. Blok, G. H. Visser, P. Govaert, and S. Van Huffel, "Neonatal seizure localization using PARAFAC decomposition," *Clinical Neurophysiology*, vol. 120, no. 10, pp. 1787–1796, 2009.

[190] T. Lin and S. Bourennane, "Hyperspectral image processing by jointly filtering wavelet component tensor," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 6, pp. 3529–3541, 2013.

[191] G. Easley, D. Labate, and W.-Q. Lim, "Sparse directional image representations using the discrete shearlet transform," *Applied and Computational Harmonic Analysis*, vol. 25, no. 1, pp. 25–46, 2008.

[192] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, 1998, pp. 903–995.

[193] N. E. Huang and S. S. Shen, *Hilbert–Huang transform and its applications*. World Scientific, 2014, vol. 16.

[194] M. Smith and T. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 3, pp. 434–441, 1986.

[195] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proceedings of the IEEE*, vol. 78, no. 1, pp. 56–93, 1990.

[196] G. Strang and T. Nguyen, *Wavelets and filter banks*. SIAM, 1996.

[197] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan 2003.

[198] H. N. Bharath, D. M. Sima, N. Sauwen, U. Himmelreich, L. De Lathauwer, and S. Van Huffel, "Non-negative canonical polyadic decomposition for tissue type differentiation in gliomas," *IEEE Journal of Biomedical and Health Informatics*, 2016.

[199] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[200] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[201] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[202] C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1 – 10, 1991.

[203] P. Comon and C. Jutten, *Handbook of blind source separation: Independent component analysis and applications*. Academic Press, 2010.

[204] A. Belouchrani, M. G. Amin, N. Thirion-Moreau, and Y. D. Zhang, "Source separation and localization using time–frequency distributions: An overview," *IEEE Signal Processing Magazine*, vol. 30, no. 6, pp. 97–107, 2013.

[205] L. Tong, V. Soon, Y. Huang, and R. Liu, "AMUSE: A new blind identification algorithm," in *International Symposium on Circuits and Systems*. IEEE, 1990, pp. 1784–1787.

[206] A. Belouchrani, K. Abed-Meraim, J. Cardoso, and E. Moulines, "Second-order blind separation of temporally correlated sources," in *Proceedings of the International Conference on Digital Signal Processing*. Citeseer, 1993, pp. 346–351.

[207] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of nonstationary sources," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1837–1848, 2001.

[208] A. Souloumiac, "Blind source detection and separation using second order non-stationarity," in *Proceedings of the IEEE International Con-*

ference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, 1995, pp. 1912–1915.

[209] Y. Shen, B. Baingana, and G. B. Giannakis, "Inferring directed network topologies via tensor factorization," in *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, Nov 2016, pp. 1739–1743.

[210] L. De Lathauwer and B. De Moor, "On the blind separation of non-circular sources," in *Proceedings of the 11th IEEE European Signal Processing Conference (EUSIPCO)*, 2002, pp. 1–4.

[211] A. Belouchrani and M. G. Amin, "Blind source separation based on time–frequency signal representations," *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 2888–2897, 1998.

[212] O. Friman, J. Cedefamn, P. Lundberg, M. Borga, and H. Knutsson, "Detection of neural activity in functional MRI using canonical correlation analysis," *Magnetic Resonance in Medicine*, vol. 45, no. 2, pp. 323–330, 2001.

[213] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[214] A. Yeredor, "Blind source separation via the second characteristic function," *Signal Processing*, vol. 80, no. 5, pp. 897–902, 2000.

[215] P. Comon and M. Rajih, "Blind identification of under-determined mixtures based on the characteristic function," *Signal Processing*, vol. 86, no. 9, pp. 2271–2281, 2006.

[216] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[217] J. Treichler and B. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 2, pp. 459–472, 1983.

[218] J.-F. Cardoso, "Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1991, pp. 3109–3112.

[219] O. Debals, M. Sohail, and L. De Lathauwer, "Analytical multi-modulus algorithms based on coupled canonical polyadic decompositions," September 2016, Technical Report 16–150, ESAT–STADIUS, KU Leuven, Belgium.

[220] O. Grellier and P. Comon, "Analytical blind discrete source separation," in *Proceedings of the 10th IEEE European Signal Processing Conference (EUSIPCO)*, Sept 2000, pp. 1–4.

[221] V. Zarzoso and P. Comon, "Blind and semi-blind equalization based on the constant power criterion," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4363–4375, 2005.

[222] A.-J. van der Veen, "Analytical method for blind binary signal separation," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 1078–1082, 1997.

[223] A.-J. van der Veen and J. Tol, "Separation of zero/constant modulus signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, 1997, pp. 3445–3448.

[224] C. L. Nikias and A. P. Petropulu, *Higher-order spectra analysis: A nonlinear signal processing framework*. PTR Prentice Hall, Englewood Cliffs, NJ, 1993.

[225] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[226] C. L. Nikias and J. M. Mendel, "Signal processing with higher-order spectra," *IEEE Signal Processing Magazine*, vol. 10, no. 3, pp. 10–37, 1993.

[227] P. McCullagh, *Tensor methods in statistics*. Chapman and Hall London, 1987, vol. 161.

[228] E. G. Thomas, "A polarization identity for multilinear maps," *Indagationes Mathematicae*, vol. 25, no. 3, pp. 468–474, 2014.

[229] T. N. Thiele, "The theory of observations," *The Annals of Mathematical Statistics*, no. 2, pp. 165–308, 1931.

[230] J. Bennett, *Collected Papers of R.A. Fisher*. University of Adelaide Press, 1972, vol. 2.

[231] J. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications," *Proceedings of the IEEE*, vol. 79, no. 3, pp. 278–305, Mar 1991.

[232] F. Kenney and E. Keeping, *Mathematics of statistics – Part Two*. D. Van Nostrand Company, Inc., 1951.

[233] J.-F. Cardoso, "Source separation using higher order moments," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1989, pp. 2109–2112.

[234] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A prewhitening-induced bound on the identification error in independent component analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 3, pp. 546–554, 2005.

[235] C. Bourin and P. Bondon, "Efficiency of high-order moment estimates," *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 255–258, 1998.

[236] J. A. R. Fonollosa, "Sample cumulants of stationary processes: Asymptotic results," *IEEE Transactions on Signal Processing*, vol. 43, no. 4, pp. 967–977, 1995.

[237] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, 2002.

[238] P. Tichavský and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 878–891, 2009.

[239] J.-F. Cardoso, S. Bose, and B. Friedlander, "Output cumulant matching for source separation," in *IEEE Signal Processing / ATHOS Workshop on Higher-Order Statistics*, 1995, pp. 44–50.

[240] A. Yeredor, "Blind separation of Gaussian sources via second-order statistics with asymptotically optimal weighting," *IEEE Signal Processing Letters*, vol. 7, no. 7, pp. 197–200, 2000.

[241] E. Eidinger and A. Yeredor, "Blind source separation via the second characteristic function with asymptotically optimal weighting," in *Proceedings of the 23rd IEEE Convention of Electrical and Electronics Engineers*, 2004, pp. 404–407.

[242] K. Thulasiraman and M. N. Swamy, *Graphs: Theory and algorithms*. John Wiley & Sons, 2011.

[243] T. Wu, A. R. Benson, and D. F. Gleich, "General tensor spectral co-clustering for higher-order data," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 29, 2016, pp. 2559–2567.

[244] T. G. Kolda, B. W. Bader, and J. P. Kenny, "Higher-order web link analysis using multilinear algebra," in *Proceedings of the 5th IEEE International Conference on Data Mining*, 2005.

[245] E. E. Papalexakis, L. Akoglu, and D. Ience, "Do more views of a graph help? Community detection and clustering in multi-graphs," in *Proceedings of the 16th IEEE International Conference on Information fusion (FUSION)*, 2013, pp. 899–905.

[246] E. Al-Sharoa, M. Al-khassaweneh, and S. Aviyente, "A tensor-based framework for community detection in dynamic networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2017, pp. 2312–2316.

[247] M. Vandecappelle, M. Boussé, F. Van Eeghem, and L. De Lathauwer, "Tensor decompositions for graph clustering," 2016, Technical Report 16–170, ESAT–STADIUS, KU Leuven, Belgium.

[248] Y. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "MetaFac: Community discovery via relational hypergraph factorization," in *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2009, pp. 527–536.

[249] A. R. Benson, D. F. Gleich, and J. Leskovec, "Tensor spectral clustering for partitioning higher-order network structures," in *Proceedings of the SIAM International Conference on Data Mining*, 2015, pp. 118–126.

[250] D. F. Gleich, L.-H. Lim, and Y. Yu, "Multilinear pagerank," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 4, pp. 1507–1541, 2015.

[251] A. Anandkumar, R. Ge, D. Hsu, and S. Kakade, "A tensor spectral approach to learning mixed membership community models," in *Proceedings of the Conference on Learning Theory*, 2013, pp. 867–881.

[252] M. Janzamin, H. Sedghi, and A. Anandkumar, "Score function features for discriminative learning: Matrix and tensor framework," *arXiv preprint arXiv:1412.2863*, 2014.

[253] S. Haykin, *Neural networks: A comprehensive foundation*, 2nd ed. Prentice Hall, 2004.

[254] J. Liang and Z. Ding, "Blind MIMO system identification based on cumulant subspace decomposition," *IEEE Transactions on Signal Processing*, vol. 51, no. 6, pp. 1457–1468, 2003.

[255] C. Fernandes, G. Favier, J. Mota *et al.*, "PARAFAC-based blind identification of convolutive MIMO linear systems," in *System Identification*, vol. 15, no. 1, 2009, pp. 1704–1709.

[256] F. Van Eeghem, M. Sørensen, and L. De Lathauwer, "Tensor decompositions with several block-Hankel factors and application in blind system identification," *IEEE Transactions on Signal Processing*, April 2017, To appear.

[257] M. De Vos, D. Nion, S. Van Huffel, and L. De Lathauwer, "A combination of parallel factor and independent component analysis," *Signal Processing*, vol. 92, no. 12, pp. 2990–2999, 2012.



Otto Debals obtained the M.Sc. degree in mathematical engineering from KU Leuven, Belgium, in 2013. He is a Ph.D. degree candidate affiliated with the Group Science, Engineering and Technology of Kulak, KU Leuven and with the STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics of the Electrical Engineering Department (ESAT), KU Leuven. His research concerns the tensorization of vector and matrix data, with further interests in tensor decompositions, optimization, signal processing, blind signal separation and blind system identification.



Lieven De Lathauwer received the Ph.D. degree from the Faculty of Engineering, KU Leuven, Belgium, in 1997. From 2000 to 2007 he was Research Associate with the Centre National de la Recherche Scientifique, France. He is currently Professor with KU Leuven. He is affiliated with both the Group Science, Engineering and Technology of Kulak, KU Leuven and with the Stadius Center for Dynamical Systems, Signal Processing and Data Analytics of the Electrical Engineering Department (ESAT), KU Leuven. He is Associate Editor of the SIAM Journal on Matrix Analysis and Applications and has served as Associate Editor for the IEEE Transactions on Signal Processing. He is Fellow of the IEEE and Fellow of SIAM. His research concerns the development of tensor tools for engineering applications.