Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations Part 2 Applications and Future Perspectives

Andrzej Cichocki

Anh-Huy Phan

Qibin Zhao

Namgil Lee

Ivan Oseledets

Masashi Sugiyama

Danilo Mandic

the essence of knowledge

Boston — Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

A. Cichocki *et al.*. *Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations Part 2 Applications and Future Perspectives*. Foundations and Trends® in Machine Learning, vol. 9, no. 6, pp. 431–673, 2016.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-276-1

Foundations and Trends® in Machine Learning
Volume 9, Issue 6, 2016

Editorial Scope

Topics

Foundations and Trends® in Machine Learning publishes survey and tutorial articles on the theory, algorithms and applications of machine learning, including the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive, and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians

Foundations and Trends® in Machine Learning, 2016, Volume 9, 6 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Foundations and Trends® in Machine Learning
Vol. 9, No. 6 (2016) 431–673

DOI: 10.1561/2200000067

Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations Part 2 Applications and Future Perspectives

Andrzej Cichocki
Riken, Brain Science Institute, Japan
Skolkovo Institute of Science and Technology (Skoltech), Russia
Systems Research Institute, Polish Academy of Science, Poland
a.cichocki@riken.jp

Anh-Huy Phan
Riken, Brain Science Institute, Japan
phan@brain.riken.jp

Qibin Zhao
Riken Center for Advanced Intelligence Project, Japan
qibin.zhao@riken.jp

Namgil Lee
Kangwon National University, Republic of Korea
namgil.lee@kangwon.ac.kr

Ivan Oseledets
Skolkovo Institute of Science and Technology (Skoltech), Russia
Institute of Numerical Mathematics of Russian Academy of Science
i.oseledets@skolkovotech.ru

Masashi Sugiyama
Riken Center for Advanced Intelligence Project, Japan
University of Tokyo
sugi@k.u-tokyo.ac.jp

Danilo Mandic
Imperial College, Department of Electrical and Electronic Engineering, UK
d.mandic@imperial.ac.uk

Contents

Abstract

Part 2 of this monograph builds on the introduction to tensor networks and their operations presented in Part 1. It focuses on tensor network models for super-compressed higher-order representation of data/parameters and related cost functions, while providing an outline of their applications in machine learning and data analytics.

A particular emphasis is on the tensor train (TT) and Hierarchical Tucker (HT) decompositions, and their physically meaningful interpretations which reflect the scalability of the tensor network approach. Through a graphical approach, we also elucidate how, by virtue of the underlying low-rank tensor approximations and sophisticated contractions of core tensors, tensor networks have the ability to perform distributed computations on otherwise prohibitively large volumes of data/parameters, thereby alleviating or even eliminating the curse of dimensionality.

The usefulness of this concept is illustrated over a number of applied areas, including generalized regression and classification (support tensor machines, canonical correlation analysis, higher order partial least squares), generalized eigenvalue decomposition, Riemannian optimization, and in the optimization of deep neural networks.

Part 1 and Part 2 of this work can be used either as stand-alone separate texts, or indeed as a conjoint comprehensive review of the exciting field of low-rank tensor networks and tensor decompositions.

A. Cichocki *et al. Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Part 2 Potential Applications and Perspectives*. Foundations and Trends® in Machine Learning, vol. 9, no. 6, pp. 431–673, 2016.
DOI: 10.1561/2200000067.

1

Tensorization and Structured Tensors

The concept of *tensorization* refers to the generation of higher-order structured tensors from the lower-order data formats (e.g., vectors, matrices or even low-order tensors), or the representation of very large scale system parameters in low-rank tensor formats. This is an essential step prior to multiway data analysis, unless the data itself is already collected in a multiway format; examples include color image sequences where the R, G and B frames are stacked into a 3rd-order tensor, or multichannel EEG signals combined into a tensor with modes, e.g., channel $\times$ time $\times$ epoch. For any given original data format, the tensorization procedure may affect the choice and performance of a tensor decomposition in the next stage.

Entries of the so constructed tensor can be obtained through: *i*) a particular rearrangement, e.g., reshaping of the original data to a tensor, *ii*) alignment of data blocks or epochs, e.g., slices of a third-order tensor are epochs of multi-channel EEG signals, or *iii*) data augmentation through, e.g., Toeplitz and Hankel matrices/tensors. In addition, tensorization of fibers of a lower-order tensor will yield a tensor of higher order. A tensor can also be generated using transform-domain methods, for example, by a time-frequency transformation via the short time Fourier transform or wavelet transform. The latter procedure is most common for multichannel data, such as EEG, where, e.g., S channels of EEG are recorded over T time samples, to produce S matrices of $F \times T$ dimensional time-frequency spectrograms stacked together into an $F \times T \times S$ dimensional third-order tensor. A tensor can also represent the data at multi-scale and orientation levels by using, e.g., the Gabor, countourlet, or pyramid steerable transformations. When exploiting statistical independence of latent variables, tensors can be generated by means of higher-order statistics (cumulants) or by partial derivatives of the Generalised Characteristic Functions (GCF) of the observations. Such tensors are usually partially or fully symmetric, and their entries represent mutual interaction between latent variables. This kind of tensorization is commonly used in ICA, BSS and blind identification of a mixing matrix. In a similar way, a symmetric tensor can be generated through measures of distances between observed entities, or their information exchange. For example, a third-order tensor, created to analyse common structures spread over EEG channels, can comprise distance matrices of pair-wise correlation or other metrics, such as causality over trials. A symmetric third-order tensor can involve three-way similarities. For such a tensorization, symmetric tensor decompositions with nonnegativity constraints are particularly well-suited.

Tensorization can also be performed through a suitable representation of the estimated parameters in some low-rank tensor network formats. This method is often used when the number of estimated parameters is huge, e.g., in modelling system response in a nonlinear system, in learning weights in a deep learning network. In this way, computation on the parameters, e.g., multiplication, convolution, inner product, Fourier transform, can be performed through core tensors of smaller scale.

One of the main motivations to develop various types of tensorization is to take advantage of data super-compression inherent in tensor network formats, especially in quantized tensor train (QTT) formats. In general, the type of tensorization depends on a specific task in hand and the structure presented in data. The next sections introduce some common tensorization methods employed in blind source separation, harmonic retrieval, system identification, multivariate polynomial regression, and nonlinear feature extraction.

1.1 Reshaping or Folding

The simplest way of tensorization is through the reshaping or folding operations, also known as segmentation (Debals and De Lathauwer, 2015; Boussé *et al.*, 2015). This type of tensorization preserves the number of original data entries and their sequential ordering, as it only rearranges a vector to a matrix or tensor. Hence, folding does not require additional memory space.

Folding. A tensor $\underline{\mathbf{Y}}$ of size $I_1 \times I_2 \times \cdots \times I_N$ is considered a folding of a vector $\mathbf{y}$ of length $I_1 I_2 \cdots I_N$, if

$$\underline{\mathbf{Y}}(i_1, i_2, \ldots, i_N) = \mathbf{y}(i)\,, \tag{1.1}$$

for all $1 \leqslant i_n \leqslant I_n$, where $i = 1 + \sum_{n=1}^{N}(i_n - 1)\prod_{k=1}^{n-1} I_k$ is a linear index of $(i_1, i_2, \ldots, i_2)$.

In other words, the vector $\mathbf{y}$ is vectorization of the tensor $\underline{\mathbf{Y}}$, while $\underline{\mathbf{Y}}$ is a tensorization of $\mathbf{y}$.

As an example, the arrangement of elements in a matrix of size $I \times L/I$, which is folded from a vector $\mathbf{y}$ of length L is given by

$$\mathbf{Y} = \begin{bmatrix} y(1) & y(I+1) & \cdots & y(L-I+1) \\ y(2) & y(I+2) & \cdots & y(L-I+2) \\ \vdots & \vdots & \ddots & \vdots \\ y(I) & y(2I) & \cdots & y(L) \end{bmatrix}. \tag{1.2}$$

Higher-order folding/reshaping refers to the application of the folding procedure several times, whereby a vector $\mathbf{y} \in \mathbb{R}^{I_1 I_2 \cdots I_N}$ is converted into an Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$.

Application to BSS. It is important to notice that a higher-order folding (quantization) of a vector of length q^N $(q = 2, 3, \ldots)$, sampled from an exponential function $y_k = a z^{k-1}$, yields an Nth-order tensor of rank 1. Moreover, wide classes of functions formed by products and/or sums of trigonometric, polynomial and rational functions can be quantized in this way to yield (approximate) low-rank tensor train (TT) network formats (Khoromskij, 2011a,b; Oseledets, 2012). Exploitation of such low-rank representations allows us to separate the signals from a single or a few mixtures, as outlined below.

Consider a single mixture, $y(t)$, which is composed of J component signals, $x_j(t)$, $j = 1, \ldots, J$, and corrupted by additive Gaussian noise, $n(t)$, to give

$$y(t) = a_1 x_1(t) + a_2 x_2(t) + \cdots + a_J x_J(t) + n(t). \tag{1.3}$$

The aim is to extract the unknown sources (components) $x_j(t)$ from the observed signal $y(t)$. Assume that higher-order foldings, $\underline{\mathbf{X}}_j$, of the component signals, $x_j(t)$, have low-rank representations in, e.g., the CP or Tucker format, given by

$$\underline{\mathbf{X}}_j = [\![\underline{\mathbf{G}}_j; \mathbf{U}_j^{(1)}, \mathbf{U}_j^{(2)}, \ldots, \mathbf{U}_j^{(N)}]\!]\,,$$

or in the TT format

$$\underline{\mathbf{X}}_j = \langle\!\langle \underline{\mathbf{G}}_j^{(1)}, \underline{\mathbf{G}}_j^{(2)}, \ldots, \underline{\mathbf{G}}_j^{(N)} \rangle\!\rangle,$$

or in any other tensor network format. Because of the multi-linearity of this tensorization, the following relation between the tensorization of the mixture, $\underline{\mathbf{Y}}$, and the tensorization of the hidden components, $\underline{\mathbf{X}}_j$, holds

$$\underline{\mathbf{Y}} = a_1 \underline{\mathbf{X}}_1 + a_2 \underline{\mathbf{X}}_2 + \cdots + a_J \underline{\mathbf{X}}_J + \underline{\mathbf{N}}\,, \tag{1.4}$$

where $\underline{\mathbf{N}}$ is the tensorization of the noise $n(t)$.

Now, by a decomposition of $\underline{\mathbf{Y}}$ into J blocks of tensor networks, each corresponding to a tensor network (TN) representation of a hidden component signal, we can find approximations of $\underline{\mathbf{X}}_j$ and the separate component signals up to a scaling ambiguity. The separation method can be used in conjunction with the Toeplitz and Hankel foldings. Example 1.9 illustrates the separation of damped sinusoid signals.

1.2 Tensorization through a Toeplitz/Hankel Tensor

1.2.1 Toeplitz Folding

The Toeplitz matrix is a structured matrix with constant entries in each diagonal. Toeplitz matrices appear in many signal processing applications, e.g., through covariance matrices in prediction, estimation, detection, classification, regression, harmonic analysis, speech enhancement, interference cancellation, image restoration, adaptive filtering, blind deconvolution and blind equalization (Bini, 1995; Gray, 2006).

Before introducing a generalization of a Toeplitz matrix to a Toeplitz tensor, we shall first consider the discrete convolution between two vectors $\mathbf{x}$ and $\mathbf{y}$ of respective lengths I and $L > I$, given by

$$\mathbf{z} = \mathbf{x} * \mathbf{y} \,. \tag{1.5}$$

Now, we can write the entries $\mathbf{z}_{I:L} = [z(I), z(I+1), \ldots, z(L)]^{\mathrm{T}}$ in a linear algebraic form as

$$\begin{aligned}\mathbf{z}_{I:L} &= \begin{bmatrix} y(I) & y(I-1) & y(I-2) & \cdots & y(1) \\ y(I+1) & y(I) & y(I-1) & \cdots & y(2) \\ y(I+2) & y(I+1) & y(I) & \cdots & y(3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(L) & y(L-1) & y(L-2) & \cdots & y(J) \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(I) \end{bmatrix} \\ &= \mathbf{Y}^{\mathrm{T}}\mathbf{x} = \mathbf{Y} \,\bar{\times}_1 \mathbf{x},\end{aligned}$$

where $J = L - I + 1$. With this representation, the convolution can be computed through a linear matrix operator, $\mathbf{Y}$, which is called the *Toeplitz matrix* of the generating vector $\mathbf{y}$.

Toeplitz matrix. A Toeplitz matrix of size $I \times J$, which is constructed from a vector $\mathbf{y}$ of length $L = I + J - 1$, is defined as

$$\mathbf{Y} = \mathcal{T}_{I,J}(\mathbf{y}) = \begin{bmatrix} y(I) & y(I+1) & \cdots & y(L) \\ y(I-1) & y(I) & \cdots & y(L-1) \\ \vdots & \vdots & \ddots & \vdots \\ y(1) & y(2) & \cdots & y(L-I+1) \end{bmatrix} . \tag{1.6}$$

The first column and first row of the Toeplitz matrix represent its entire generating vector.

Indeed, all $(L + I - 1)$ entries of $\mathbf{y}$ in the above convolution (1.5) can be expressed either by: (i) using a Toeplitz matrix formed from a zero-padded generating vector $[\mathbf{0}_{I-1}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \mathbf{0}_{I-1}^{\mathrm{T}}]^{\mathrm{T}}$, with $[\mathbf{y}^{\mathrm{T}}, \mathbf{0}_{I-1}^{\mathrm{T}}]$ being the first row of this Toeplitz matrix, to give

$$\mathbf{z} = \mathcal{T}_{I,L+I-1}([\mathbf{0}_{I-1}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \mathbf{0}_{I-1}^{\mathrm{T}}]^{\mathrm{T}})^{\mathrm{T}}\, \mathbf{x} \,, \tag{1.7}$$

or (ii) through a Toeplitz matrix of the generating vector $[\mathbf{0}_{L-1}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{0}_{L-1}^{\mathrm{T}}]^{\mathrm{T}}$, to yield

$$\mathbf{z} = \mathcal{T}_{L,L+I-1}([\mathbf{0}_{L-1}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{0}_{L-1}^{\mathrm{T}}]^{\mathrm{T}})^{\mathrm{T}}\, \mathbf{y}\,. \tag{1.8}$$

The so expanded Toeplitz matrix is a circulant matrix of $[\mathbf{y}^{\mathrm{T}}, \mathbf{0}_{I-1}^{\mathrm{T}}]^{\mathrm{T}}$.

Consider now a convolution of three vectors, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{y}$ of respective lengths I_1, I_2 and $(L \geqslant I_1 + I_2)$, given by

$$\mathbf{z} = \mathbf{x}_1 * \mathbf{x}_2 * \mathbf{y}\,.$$

For its implementation, we first construct a Toeplitz matrix, $\mathbf{Y}$, of size $I_1 \times (L - I_1 + 1)$ from the generating vector $\mathbf{y}$. Then, we use the rows $\mathbf{Y}(k,:)$ to generate Toeplitz matrices, $\mathbf{Y}_k$ of size $I_2 \times I_3$. Finally, all I_1 Toeplitz matrices, $\mathbf{Y}_1$, ..., $\mathbf{Y}_{I_1}$, are stacked as horizontal slices of a third-order tensor $\underline{\mathbf{Y}}$, i.e., $\underline{\mathbf{Y}}(k,:,:) = \mathbf{Y}_k$, $k = 1, \ldots, I_1$. It can be verified that entries $[z(I_1 + I_2 - 1), \ldots, z(L)]^T$ can be computed as

$$\begin{bmatrix} z(I_1 + I_2 - 1) \\ \vdots \\ z(L) \end{bmatrix} = [\mathbf{x}_1 * \mathbf{x}_2 * \mathbf{y}]_{I_1+I_2-1:L} = \underline{\mathbf{Y}} \,\bar{\times}_1\, \mathbf{x}_1 \,\bar{\times}_2\, \mathbf{x}_2.$$

The tensor $\underline{\mathbf{Y}}$ is referred to as the Toeplitz tensor of the generating vector $\mathbf{y}$.

Toeplitz tensor. An Nth-order Toeplitz tensor of size $I_1 \times I_2 \times \cdots \times I_N$, which is represented by $\underline{\mathbf{Y}} = \mathcal{T}_{I_1,\ldots,I_N}(\mathbf{y})$, is constructed from a generating vector $\mathbf{y}$ of length $L = I_1 + I_2 + \cdots + I_N - N + 1$, such that its entries are defined as

$$\underline{\mathbf{Y}}(i_1, \ldots, i_{N-1}, i_N) = y(\bar{i}_1 + \cdots + \bar{i}_{N-1} + i_N)\,, \tag{1.9}$$

where $\bar{i}_n = I_n - i_n$. An example of the Toeplitz tensor is illustrated in Figure 1.1.

Example 1.1. Given a $3 \times 3 \times 3$ dimensional Toeplitz tensor of a sequence $1, 2, \ldots, 7$, the horizontal slices are Toeplitz matrices of sizes

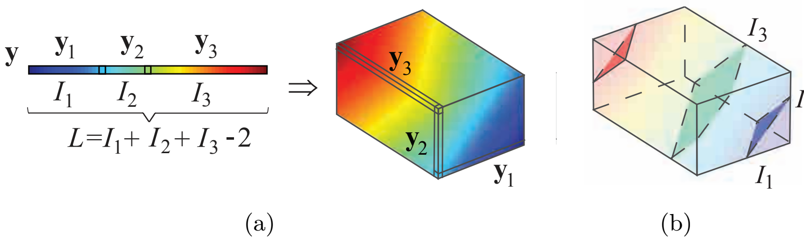


Figure 1.1: Illustration of a 3rd-order Toeplitz tensor of size $I_1 \times I_2 \times I_3$, generated from a vector $\mathbf{y}$ of length $L = I_1 + I_2 + I_3 - 2$. (a) The highlighted fibers of the Toeplitz tensor form the generating vector $\mathbf{y}$. (b) The entries in every shaded diagonal intersection are identical and represent one element of $\mathbf{y}$.

3×3 given by

$$\mathcal{T}_{3,3,3}(1,\ldots,7) = \begin{bmatrix} \mathcal{T}_{3,3}(3,\ldots,7) \\ \mathcal{T}_{3,3}(2,\ldots,6) \\ \mathcal{T}_{3,3}(1,\ldots,5) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 5 & 6 & 7 \\ 4 & 5 & 6 \\ 3 & 4 & 5 \end{bmatrix} \\ \begin{bmatrix} 4 & 5 & 6 \\ 3 & 4 & 5 \\ 2 & 3 & 4 \end{bmatrix} \\ \begin{bmatrix} 3 & 4 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 3 \end{bmatrix} \end{bmatrix}.$$

Recursive generation. An Nth-order Toeplitz tensor of a generating vector $\mathbf{y}$ is of size $I_1 \times I_2 \times \cdots \times I_N$, can be constructed from an $(N-1)$th-order Toeplitz tensor of size $I_1 \times I_2 \times \cdots \times (I_{N-1} + I_N - 1)$ of the same generating vector, by a conversion of mode-$(N-1)$ fibers to Toeplitz matrices of size $I_{N-1} \times I_N$.

Following the definition of the Toeplitz tensor, the convolution of $(N-1)$ vectors, $\mathbf{x}_n$ of respective lengths I_n, and a vector $\mathbf{y}$ of length L, can be represented as a tensor-vector product of an Nth-order Toeplitz tensor and vectors $\mathbf{x}_n$, that is

$$[\mathbf{x}_1 * \mathbf{x}_2 * \cdots * \mathbf{x}_{N-1} * \mathbf{y}]_{J:L} = \underline{\mathbf{Y}} \,\bar{\times}_1\, \mathbf{x}_1 \,\bar{\times}_2\, \mathbf{x}_2 \cdots \bar{\times}_{N-1}\, \mathbf{x}_{N-1}\,,$$

where $\underline{\mathbf{Y}} = \mathcal{T}_{I_1,\ldots,I_{N-1},L-J}(\mathbf{y})$ is a Toeplitz tensor of size $I_1 \times \cdots \times I_{N-1} \times (L-J)$ generated from $\mathbf{y}$, and $J = \sum_{n=1}^{N-1} I_n - N + 1$, or

$$\mathbf{x}_1 * \mathbf{x}_2 * \cdots * \mathbf{x}_{N-1} * \mathbf{y} = \underline{\widetilde{\mathbf{Y}}} \,\bar{\times}_1\, \mathbf{x}_1 \,\bar{\times}_2\, \mathbf{x}_2 \cdots \bar{\times}_{N-1}\, \mathbf{x}_{N-1}\,,$$

where $\underline{\widetilde{\mathbf{Y}}} = \mathcal{T}_{I_1,\ldots,I_{N-1},L+J}([\mathbf{0}_J^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \mathbf{0}_J^{\mathrm{T}}]^{\mathrm{T}})$ is a Toeplitz tensor, of the zero-padded vector of $\mathbf{y}$, is of size $I_1 \times \cdots \times I_{N-1} \times (L+J)$.

1.2.2 Hankel Folding

The Hankel matrix and Hankel tensor have similar structures to the Toeplitz matrix and tensor and can also be used as linear operators in the convolution.

Hankel matrix. An $I \times J$ Hankel matrix of a vector $\mathbf{y}$, of length $L = I + J - 1$, is defined as

$$\mathbf{Y} = \mathcal{H}_{I,J}(\mathbf{y}) = \begin{bmatrix} y(1) & y(2) & \cdots & y(J) \\ y(2) & y(3) & \cdots & y(J+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(I) & y(I+1) & \cdots & y(L) \end{bmatrix}. \qquad (1.10)$$

Hankel tensor. (Papy *et al.*, 2005) An Nth-order Hankel tensor of size $I_1 \times I_2 \times \cdots \times I_N$, which is represented by $\underline{\mathbf{Y}} = \mathcal{H}_{I_1,\ldots,I_N}(\mathbf{y})$, is constructed from a generating vector $\mathbf{y}$ of length $L = \sum_n I_n - N + 1$, such that its entries are defined as

$$\underline{\mathbf{Y}}(i_1, i_2, \ldots, i_N) = y(i_1 + i_2 + \cdots + i_N - N + 1)\,. \qquad (1.11)$$

Remark 1.1. (Properties of a Hankel tensor)

- The generating vector $\mathbf{y}$ can be reconstructed by a concatenation of fibers of the Hankel tensor $\underline{\mathbf{Y}}(I_1, \ldots, I_{n-1}, :, 1, \ldots, 1)$, where $n = 1, \ldots, N-1$, and

$$\mathbf{y} = \begin{bmatrix} \underline{\mathbf{Y}}(1 : I_1 - 1, 1, \ldots, 1) \\ \vdots \\ \underline{\mathbf{Y}}(I_1, \ldots, I_{n-1}, 1 : I_n - 1, 1, \ldots, 1) \\ \vdots \\ \underline{\mathbf{Y}}(I_1, \ldots, I_{N-1}, 1 : I_N) \end{bmatrix}. \qquad (1.12)$$

- Slices of a Hankel tensor $\underline{\mathbf{Y}}$, i.e., any subset of the tensor produced by fixing $(N-2)$ indices of its entries and varying the two remaining indices, are also Hankel matrices.

- An Nth-order Hankel tensor, $\mathcal{H}_{I_1,\ldots,I_{N-1},I_N}(\mathbf{y})$, can be constructed from an $(N-1)$th-order Hankel tensor $\mathcal{H}_{I_1,\ldots,I_{N-2},I_{N-1}+I_N-1}(\mathbf{y})$ of size $I_1 \times \cdots \times I_{N-2} \times (I_{N-1}+I_N-1)$ by converting its mode-$(N-1)$ fibers to Hankel matrices of size $I_{N-1} \times I_N$.

- Similarly to the Toeplitz tensor, the convolution of $(N-1)$ vectors, $\mathbf{x}_n$ of lengths I_n, and a vector $\mathbf{y}$ of length L, can be represented as

$$[\mathbf{x}_1 * \mathbf{x}_2 * \cdots * \mathbf{x}_{N-1} * \mathbf{y}]_{J:L} = \underline{\mathbf{Y}} \,\bar{\times}_1\, \tilde{\mathbf{x}}_1 \,\bar{\times}_2\, \tilde{\mathbf{x}}_2 \cdots \bar{\times}_{N-1}\, \tilde{\mathbf{x}}_{N-1},$$

or

$$\mathbf{x}_1 * \mathbf{x}_2 * \cdots * \mathbf{x}_{N-1} * \mathbf{y} = \underline{\widetilde{\mathbf{Y}}} \,\bar{\times}_1\, \tilde{\mathbf{x}}_1 \,\bar{\times}_2\, \tilde{\mathbf{x}}_2 \cdots \bar{\times}_{N-1}\, \tilde{\mathbf{x}}_{N-1},$$

where $\tilde{\mathbf{x}}_n = [x_n(I_n), \ldots, x_n(2), x_n(1)]$, $J = \sum_n I_n - N + 1$, $\underline{\mathbf{Y}} = \mathcal{H}_{I_1,\ldots,I_{N-1},L-J}(\mathbf{y})$ is the Nth-order Hankel tensor of $\mathbf{y}$, whereas $\underline{\widetilde{\mathbf{Y}}} = \mathcal{H}_{I_1,\ldots,I_{N-1},L+J}([\mathbf{0}_J^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \mathbf{0}_J^{\mathrm{T}}]^{\mathrm{T}})$ is the Hankel tensor of a zero-padded version of $\mathbf{y}$.

- A Hankel tensor with identical dimensions $I_n = I$, for all n, is a symmetric tensor.

Example 1.2. A $3 \times 3 \times 3$ – dimensional Hankel tensor of a sequence $1, 2, \ldots, 7$ is a symmetric tensor, and is given by

$$\mathcal{H}_{3,3,3}(1:7) = \left[\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}, \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}, \begin{bmatrix} 3 & 4 & 5 \\ 4 & 5 & 6 \\ 5 & 6 & 7 \end{bmatrix} \right].$$

1.2.3 Quantized Tensorization

It is important to notice that the tensorizations into the Toeplitz and Hankel tensors typically enlarge the number of data samples (in the sense that the number of entries of the corresponding tensor is larger than the number of original samples). For example, when the dimensions $I_n = 2$ for all n, the so generated tensor to be a quantized tensor of order $(L-1)$, while the number of entries of a such tensor increases from the original size L to 2^{L-1}. Therefore, quantized tensorizations are suited to analyse signals of short-length, especially in multivariate autoregressive modelling.

1.2.4 Convolution Tensor

Consider again the convolution $\mathbf{x} * \mathbf{y}$ of two vectors of respective lengths I and L. We can then rewrite the expression for the entries-$(I, I+1, \ldots, L)$ as

$$[\mathbf{x} * \mathbf{y}]_{I:L} = \underline{\mathbf{C}} \,\bar{\times}_1\, \mathbf{x} \,\bar{\times}_3\, \mathbf{y}\,,$$

where $\underline{\mathbf{C}}$ is a third-order tensor of size $I \times J \times L$, $J = L - I + 1$, for which the $(l-I)$-th diagonal elements of l-th slices are ones, and the remaining entries are zeros, for $l = 1, 2, \ldots, L$. For example, the slices $\underline{\mathbf{C}}(:,:,l)$, for $l \leqslant I$, are given by

$$\underline{\mathbf{C}}(:,:,l) = \begin{array}{c} \begin{bmatrix} 0 & & & 0 \\ & & & \\ 1 & & \ddots & \\ & \ddots & & \\ 0 & & 1 & 0 \end{bmatrix} \\ \begin{array}{cccc} & & l & \end{array} \end{array}.$$

The tensor $\underline{\mathbf{C}}$ is called the *convolution tensor*. Illustration of a convolution tensor of size $I \times I \times (2I-1)$ is given in Figure 1.2.

Note that a product of this tensor with the vector $\mathbf{y}$ yields the Toeplitz matrix of the generating vector $\mathbf{y}$, which is of size $I \times J$, in the form

$$\underline{\mathbf{C}} \,\bar{\times}_3\, \mathbf{y} = \mathcal{T}_{I,J}(\mathbf{y})\,,$$

while the tensor-vector product $\underline{\mathbf{C}} \bar{\times}_1 \mathbf{x}$ yields a Toeplitz matrix of the generating vector $[\mathbf{0}_{L-I}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{0}_{J-1}^{\mathrm{T}}]^{\mathrm{T}}$, or a circulant matrix of

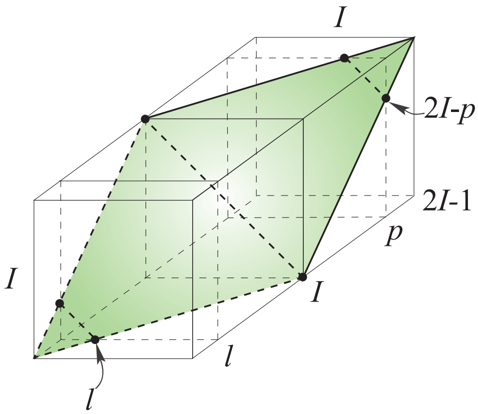


Figure 1.2: Visualization of a convolution tensor of size $I \times I \times (2I-1)$. Unit entries are located on the shaded parallelogram.

$[\mathbf{0}_{L-I}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}]^{\mathrm{T}}$

$$\underline{\mathbf{C}} \,\bar{\times}_1\, \mathbf{x} = \mathcal{T}_{L,J}([\mathbf{0}_{L-I}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{0}_{J-1}^{\mathrm{T}}]^{\mathrm{T}}) .$$

In general, for a convolution of $(N-1)$ vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}$, of respective lengths $I_1, \ldots, I_{N-1}$ and a vector $\mathbf{y}$ of length L

$$\mathbf{z} = \mathbf{x}_1 * \mathbf{x}_2 * \cdots * \mathbf{x}_{N-1} * \mathbf{y} , \tag{1.13}$$

the entries of $\mathbf{z}$ can be expressed through a multilinear product of a convolution tensor, $\underline{\mathbf{C}}$, of $(N+1)$th-order and size $I_1 \times I_2 \times \cdots \times I_N \times L$, $I_N = L - \sum_{n=1}^{N-1} I_n + N - 1$, and the N input vectors

$$\mathbf{z}_{L-I_N+1:L} = \underline{\mathbf{C}} \,\bar{\times}_1\, \mathbf{x}_1 \,\bar{\times}_2\, \mathbf{x}_2 \cdots \,\bar{\times}_{N-1}\, \mathbf{x}_{N-1} \,\bar{\times}_{N+1}\, \mathbf{y} . \tag{1.14}$$

Most entries of $\underline{\mathbf{C}}$ are zeros, except for those located at $(i_1, i_2, \ldots, i_{N+1})$, such that

$$\sum_{n=1}^{N-1} \bar{i}_n + i_N - i_{N+1} = 0 , \tag{1.15}$$

where $\bar{i}_n = I_n - i_n$, $i_n = 1, 2, \ldots, I_n$.

The tensor product $\underline{\mathbf{C}} \bar{\times}_{N+1} \mathbf{y}$ yields the Toeplitz tensor of the generating vector $\mathbf{y}$, shown below

$$\underline{\mathbf{C}} \,\bar{\times}_{N+1} \mathbf{y} = \mathcal{T}_{I_1,\ldots,I_N}(\mathbf{y}). \tag{1.16}$$

1.2.5 QTT Representation of the Convolution Tensor

An important property of the convolution tensor is that it has a QTT representation with rank no larger than the number of inputs vectors, N. To illustrate this property, for simplicity, we consider an Nth-order Toeplitz tensor of size $I \times I \times \cdots \times I$ generated from a vector of length $(N\,I - N + 1)$, where $I = 2^D$. The convolution tensor of this Toeplitz tensor is of $(N+1)$th-order and of size $I \times I \times \cdots \times I \times (N\,I - N + 1)$.
Zero-padded convolution tensor. By appending $(N-1)$ zero tensors of size $I \times I \times \cdots \times I$ before the convolution tensor, we obtain an $(N+1)$th-order convolution tensor, $\underline{\mathbf{C}}$, of size $I \times I \times \cdots \times I \times IN$.
QTT representation. The zero-padded convolution tensor can be represented in the following QTT format

$$\underline{\mathbf{C}} = \underline{\widetilde{\mathbf{C}}}^{(1)} \;|\otimes|\; \underline{\widetilde{\mathbf{C}}}^{(2)} \;|\otimes|\; \cdots \;|\otimes|\; \underline{\widetilde{\mathbf{C}}}^{(D)} \;|\otimes|\; \underline{\widetilde{\mathbf{C}}}^{(D+1)}, \tag{1.17}$$

where " $|\otimes|$ " represents the strong Kronecker product between block tensors[1] $\underline{\widetilde{\mathbf{C}}}^{(n)} = [\underline{\widetilde{\mathbf{C}}}^{(n)}_{r,s}]$ defined from the $(N+3)$th-order core tensors $\underline{\mathbf{C}}^{(n)}$ as $\underline{\widetilde{\mathbf{C}}}^{(n)}_{r,s} = \underline{\mathbf{C}}^{(n)}(r, :, \ldots, :, s)$.

The last core tensor $\underline{\mathbf{C}}^{(D+1)}$ represents an exchange (backward identity) matrix of size $N \times N$ which can represented as an $(N+3)$th-order tensor of size $N \times 1 \times \cdots \times 1 \times N \times 1$. The first D core tensors $\underline{\mathbf{C}}^{(1)}$, $\underline{\mathbf{C}}^{(2)}$, $\ldots$, $\underline{\mathbf{C}}^{(D)}$ are expressed based on the so-called elementary core tensor $\underline{\mathbf{S}}$ of size $N \times \underbrace{2 \times 2 \times \cdots \times 2}_{(N+1)\text{ dimensions}} \times N$, as

$$\underline{\mathbf{C}}^{(1)} = \underline{\mathbf{S}}(1, :, \ldots, :), \quad \underline{\mathbf{C}}^{(2)} = \cdots = \underline{\mathbf{C}}^{(D)} = \underline{\mathbf{S}}\,. \tag{1.18}$$

The rigorous definition of the elementary core tensor is provided in Appendix 5.

[1] A "block tensor" represents a multilevel matrix, the entries of which are matrices or tensors.

Table 1.1: Rank of QTT representations of convolution tensors of $(N+1)$th-order for $N = 2, \ldots, 17$.

N	QTT rank	N	QTT rank
2	2, 2, 2, ..., 2	10	6, 8, 9, ..., 9
3	2, 3, 3, ..., 3	11	6, 9, 10, ..., 10
4	3, 4, 4, ..., 4	12	7, 10, 11, ..., 11
5	3, 4, 5, ..., 5	13	7, 10, 12, ..., 12
6	4, 5, 6, ..., 6	14	8, 11, 13, ..., 13
7	4, 6, 7, ..., 7	15	8, 12, 14, ..., 14
8	5, 7, 8, ..., 8	16	9, 13, 15, ..., 15
9	5, 7, 8, ..., 8	17	9, 13, 15, ..., 15

Table 1.1 provides ranks of the QTT representation for various order of convolution tensors. The elementary core tensor $\underline{\mathbf{S}}$ can be further re-expressed in a (tensor train) TT-format with $(N+1)$ sparse TT cores, as

$$\underline{\mathbf{S}} = \langle\langle \underline{\mathbf{G}}^{(1)}, \underline{\mathbf{G}}^{(2)}, \ldots, \underline{\mathbf{G}}^{(N+1)} \rangle\rangle,$$

where $\underline{\mathbf{G}}^{(k)}$ is of size $(N+k-1) \times 2 \times (N+k)$, for $k = 1, \ldots, N$, and the last core tensor $\underline{\mathbf{G}}^{(N+1)}$ is of size $2N \times 2 \times N$.

Example 1.3. Convolution tensor of 3rd-order.

For the vectors $\mathbf{x}$ of length 2^D and $\mathbf{y}$ of length $(2^{D+1} - 1)$, the expanded convolution tensor has size of $2^D \times 2^D \times 2^{D+1}$. The elementary core tensor $\underline{\mathbf{S}}$ is then of size $2 \times 2 \times 2 \times 2 \times 2$ and its sub-tensors, $\underline{\mathbf{S}}(i, :, :, :, :)$, are given in a 2×2 block form of the last two indices through four matrices, $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathbf{S}_3$ and $\mathbf{S}_4$, of size 2×2, that is

$$\underline{\mathbf{S}}(1, :, :, :, :) = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_3 \\ \mathbf{S}_2 & \mathbf{S}_4 \end{bmatrix}, \quad \underline{\mathbf{S}}(2, :, :, :, :) = \begin{bmatrix} \mathbf{S}_2 & \mathbf{S}_4 \\ \mathbf{S}_3 & \mathbf{S}_1 \end{bmatrix},$$

where

$$\mathbf{S}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{S}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{S}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{S}_4 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

The convolution tensor can then be represented in a QTT format of rank-2 (Kazeev *et al.*, 2013) with core tensors $\underline{\mathbf{C}}^{(2)} = \cdots = \underline{\mathbf{C}}^{(D)} = \underline{\mathbf{S}}$,

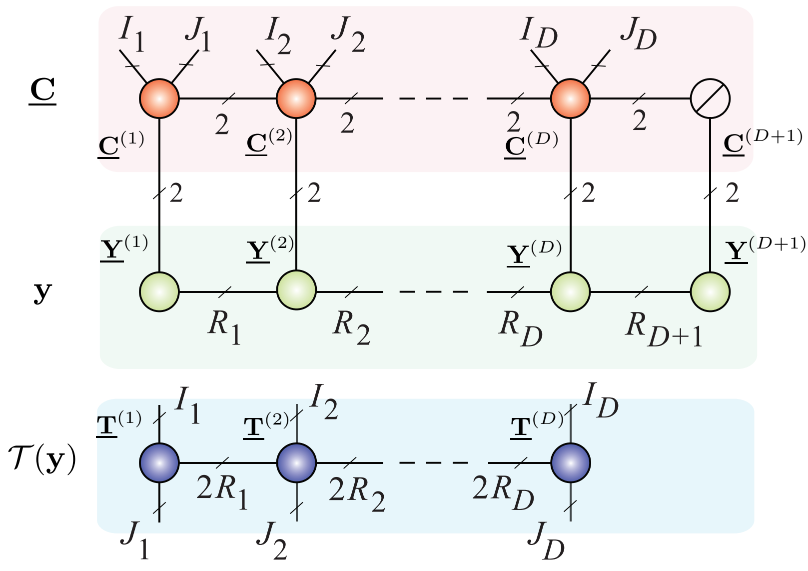


Figure 1.3: Representation of the convolution tensor in QTT format. (Top) Distributed representation of a convolution tensor $\underline{\mathbf{C}}$ of size $I \times J \times 2I$ in a QTT format, where $I = J = 2^D$. The first core tensor $\underline{\mathbf{C}}^{(1)}$ is of size $1 \times 2 \times 2 \times 2 \times 2$, the last core tensor $\underline{\mathbf{C}}^{(D+1)}$ represents a backward identity matrix, and the remaining 5th-order core tensors of size $2 \times 2 \times 2 \times 2 \times 2$ are identical. A vector $\mathbf{y}$ is of length 2^{D+1} in a QTT format. (Bottom) Generation of the Toeplitz matrix, $\mathcal{T}(\mathbf{y})$, of the vector $\mathbf{y}$ from the convolution tensor and its representation in the QTT format, $I_d = J_d = 2$ for $d = 1, \ldots, D$.

$\underline{\mathbf{C}}^{(1)} = \underline{\mathbf{S}}(1, :, :, :, :)$, and the last core tensor $\underline{\mathbf{C}}^{(D+1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ which is of size $2 \times 1 \times 1 \times 2 \times 1$. This QTT representation is useful to generate a Toeplitz matrix when its generating vector is given in the QTT format. An illustration of the convolution tensor $\underline{\mathbf{C}}$ is provided in Figure 1.3.

Example 1.4. Convolution tensor of fourth-order.

For the convolution tensor of fourth order, i.e., Toeplitz order $N = 3$, the elementary core tensor $\underline{\mathbf{S}}$ is of size $3 \times 2 \times 2 \times 2 \times 2 \times 3$, and is given in a 2×3 block form of the last two indices as

$$\underline{\mathbf{S}}(1,:,\ldots,:) = \begin{bmatrix} \underline{\mathbf{S}}_1 & \underline{\mathbf{S}}_3 & \underline{\mathbf{S}}_5 \\ \underline{\mathbf{S}}_2 & \underline{\mathbf{S}}_4 & \underline{\mathbf{S}}_6 \end{bmatrix}, \quad \underline{\mathbf{S}}(2,:,\ldots,:) = \begin{bmatrix} \underline{\mathbf{S}}_2 & \underline{\mathbf{S}}_4 & \underline{\mathbf{S}}_6 \\ \underline{\mathbf{S}}_5 & \underline{\mathbf{S}}_1 & \underline{\mathbf{S}}_3 \end{bmatrix},$$
$$\underline{\mathbf{S}}(3,:,\ldots,:) = \begin{bmatrix} \underline{\mathbf{S}}_5 & \underline{\mathbf{S}}_1 & \underline{\mathbf{S}}_3 \\ \underline{\mathbf{S}}_6 & \underline{\mathbf{S}}_2 & \underline{\mathbf{S}}_4 \end{bmatrix}.$$

where $\underline{\mathbf{S}}_n$ are of size $2 \times 2 \times 2$, $\underline{\mathbf{S}}_5$, $\underline{\mathbf{S}}_6$ are zero tensors, and

$$\underline{\mathbf{S}}_1 = \left[\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right], \quad \underline{\mathbf{S}}_2 = \left[\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\right],$$
$$\underline{\mathbf{S}}_3 = \left[\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\right], \quad \underline{\mathbf{S}}_4 = \left[\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right].$$

Finally, the zero-padded convolution tensor of size $2^D \times 2^D \times 2^D \times 3 \cdot 2^D$ has a QTT representation in (1.17) with $\underline{\mathbf{C}}^{(1)} = \underline{\mathbf{S}}(1,:,:,:,:,[1,2])$, $\underline{\mathbf{C}}^{(2)} = \underline{\mathbf{S}}([1,2],:,:,:,:,:)$, $\underline{\mathbf{C}}^{(3)} = \cdots = \underline{\mathbf{C}}^{(D)} = \underline{\mathbf{S}}$, and the last core tensor $\underline{\mathbf{C}}_{D+1} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ which is of size $3 \times 1 \times 1 \times 3 \times 1$.

1.2.6 Low-rank Representation of Hankel and Toeplitz Matrices/Tensors

The Hankel and Toeplitz foldings are multilinear tensorizations, and can be applied to the BSS problem, as in (1.4). When the Hankel and Toeplitz tensors of the hidden sources are of low-rank in some tensor network representation, the tensor of the mixture is expressed as a sum of low rank tensor terms.

For example, the Hankel and Toeplitz matrices/tensors of an exponential function, $v_k = az^{k-1}$, are rank-1 matrices/tensors, and consequently Hankel matrices/tensors of sums and/or products of exponentials, sinusoids, and polynomials will also be of low-rank, which is equal to the degree of the function being considered.

Hadamard Product. More importantly, when Hankel/Toeplitz tensors of two vectors $\mathbf{u}$ and $\mathbf{v}$ have low-rank CP/TT representations, the Hankel/Toeplitz tensor of their element-wise product, $\mathbf{w} = \mathbf{u} \circledast \mathbf{v}$, can also be represented in the same CP/TT tensor format

$$\mathcal{H}(\mathbf{u}) \circledast \mathcal{H}(\mathbf{v}) = \mathcal{H}(\mathbf{u} \circledast \mathbf{v})$$
$$\mathcal{T}(\mathbf{u}) \circledast \mathcal{T}(\mathbf{v}) = \mathcal{T}(\mathbf{u} \circledast \mathbf{v}) \,.$$

The CP/TT rank of $\mathcal{H}(\mathbf{u} \circledast \mathbf{v})$ or $\mathcal{T}(\mathbf{u} \circledast \mathbf{v})$ is not larger than the product of the CP/TT ranks of the tensors of $\mathbf{u}$ and $\mathbf{v}$.

Example 1.5. The third-order Hankel tensor of $u(t) = \sin(\omega t)$ is a rank-3 tensor, and the third-order Hankel tensor of $v(t) = t$ is of rank-2; hence the Hankel tensor of the $w(t) = t\ \sin(\omega t)$ has at most rank-6.

Symmetric CP and Vandermonde decompositions. It is important to notice that a Hankel tensor $\underline{\mathbf{Y}}$ of size $I \times I \times \cdots \times I$ can always be represented by a symmetric CP decomposition

$$\underline{\mathbf{Y}} = \underline{\mathbf{I}} \times_1 \mathbf{A} \times_2 \mathbf{A} \cdots \times_N \mathbf{A} \,.$$

Moreover, the tensor $\underline{\mathbf{Y}}$ also admits a symmetric CP decomposition with Vandermonde structured factor matrix (Qi, 2015)

$$\underline{\mathbf{Y}} = \mathrm{diag}_N(\boldsymbol{\lambda}) \times_1 \mathbf{V}^{\mathrm{T}} \times_2 \mathbf{V}^{\mathrm{T}} \cdots \times_N \mathbf{V}^{\mathrm{T}} \,, \tag{1.19}$$

where $\boldsymbol{\lambda}$ comprises R non-zero coefficients, and $\mathbf{V}$ is a Vandermonde matrix generated from R distinct values $\mathbf{v} = [v_1, v_2, \ldots, v_R]$

$$\mathbf{V} = \begin{bmatrix} 1 & v_1 & v_1^2 & \ldots & v_1^{I-1} \\ 1 & v_2 & v_2^2 & \ldots & v_2^{I-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & v_R & v_R^2 & \ldots & v_R^{I-1} \end{bmatrix} . \tag{1.20}$$

By writing the decomposition in (1.19) for the entries $\underline{\mathbf{Y}}(I_1, \ldots, I_{n-1}, :, 1, \ldots, 1)$ (see (1.12)), the Vandermonde decomposition of the Hankel tensor $\underline{\mathbf{Y}}$ becomes a Vandermonde factorization of $\mathbf{y}$ (Chen, 2016), given by

$$\mathbf{y} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ v_1 & v_2 & \ldots & v_R \\ v_1^2 & v_2^2 & \ldots & v_R^2 \\ \vdots & \vdots & \ddots & \vdots \\ v_1^{L-1} & v_2^{L-1} & \ldots & v_R^{L-1} \end{bmatrix} \boldsymbol{\lambda} \,.$$

Observe that various Vandermonde decompositions of the Hankel tensors of the same vector $\mathbf{y}$, but of different tensor orders N, have the same generating Vandermonde vector $\mathbf{v}$. Moreover, the Vandemonde rank, i.e, the minimum of R in the decomposition (1.19), therefore cannot exceed the length L of the generating vector $\mathbf{y}$.

QTT representation of Toeplitz/Hankel tensor. As mentioned previously, the zero-padded convolution tensor of $(N+1)$th-order can be represented in a QTT format of rank of at most N. Hence, if a vector $\mathbf{y}$ of length $2^D N$ has a QTT representation of rank-$(R_1, \ldots, R_D)$, given by

$$\mathbf{y} = \widetilde{\mathbf{Y}}^{(1)} \,|\!\otimes\!|\, \widetilde{\mathbf{Y}}^{(2)} \,|\!\otimes\!|\, \cdots \,|\!\otimes\!|\, \widetilde{\mathbf{Y}}^{(D+1)}\,, \tag{1.21}$$

where $\widetilde{\mathbf{Y}}^{(d)}$ is an $R_{d-1} \times R_d$ block matrix of the core tensor $\underline{\mathbf{Y}}^{(d)}$ of size $R_{d-1} \times 2 \times R_d$, for $d = 1, \ldots, D$, or of $\underline{\mathbf{Y}}^{(D+1)}$ of size $R_D \times N \times 1$, then following the relation between the convolution tensor and the Toeplitz tensor of the generating vector $\mathbf{y}$, we have

$$\mathcal{T}(\mathbf{y}) = \underline{\mathbf{C}} \,\bar{\times}_{N+1}\, \mathbf{y}\,. \tag{1.22}$$

This Nth-order Toeplitz tensor can also be represented by a QTT tensor with rank of at most $N(R_1, \ldots, R_D)$, as

$$\mathcal{T}(\mathbf{y}) = \underline{\widetilde{\mathbf{T}}}^{(1)} \,|\!\otimes\!|\, \underline{\widetilde{\mathbf{T}}}^{(2)} \,|\!\otimes\!|\, \cdots \,|\!\otimes\!|\, \underline{\widetilde{\mathbf{T}}}^{(D)}\,, \tag{1.23}$$

where $\underline{\widetilde{\mathbf{T}}}^{(d)}$ is a block tensor of the core tensor $\underline{\mathbf{T}}^{(d)}$. The core $\underline{\mathbf{T}}^{(1)}$ is of size $1 \times 2 \times \cdots \times 2 \times N\,R_1$, and cores $\underline{\mathbf{T}}^{(2)}, \ldots, \underline{\mathbf{T}}^{(D-1)}$ are of size $NR_{d-1} \times 2 \times \cdots \times 2 \times NR_d$, while the last core tensor $\underline{\mathbf{T}}^{(D)}$ is of size $NR_{D-1} \times 2 \times \cdots \times 2 \times 1$. These core tensors are core contractions between the two core tensors $\underline{\mathbf{C}}^{(d)}$ and $\underline{\mathbf{Y}}^{(d)}$. Figure 1.3 illustrates the generation of a Toeplitz matrix as a tensor-vector product of a third-order convolution tensor $\underline{\mathbf{C}}$ and a generating vector, $\mathbf{x}$, of length 2^{D+1}, both in QTT-formats. The core tensors of $\underline{\mathbf{C}}$ are given in Example 1.3.

Remarks:

- Because of zero-padding within the convolution tensor, the Toeplitz tensor of $\mathbf{y}$, generated in (1.22) and (1.23), takes only entries $\mathbf{y}(N), \mathbf{y}(N+1), \ldots, \mathbf{y}(2^D N)$, i.e., it corresponds to the Toeplitz tensor of the generating vector $\mathbf{y}(N), \mathbf{y}(N+1), \ldots, \mathbf{y}(2^D N)$.

- The Hankel tensor also admits a QTT representation in the similar form to a Toeplitz tensor (cf. (1.23)).

- Low-rank TN representation of the Toeplitz and Hankel tensors has been exploited, e.g., in blind source separation and harmonic retrieval. By verifying a low-rank TN representation of the signal in hand, we can confirm the existence of a low-rank TN representation of Toeplitz/Hankel tensors of the signal.

- QTT rank of the Toeplitz tensor in (1.23) is at most N times the QTT rank of the generating vector $\mathbf{y}$. The rank may not be minimal. For example, the sinusoid signal is of rank-2 in QTT format, and its Toeplitz tensor also has a rank-2 QTT representation.

- *Fast convolution of vectors in QTT formats.* A straightforward consequences is that when vectors $\mathbf{x}_n$ are given in their QTT formats, their convolution $\mathbf{x}_1 * \mathbf{x}_2 * \cdots * \mathbf{x}_N$ can be computed through core contractions between the core tensors of the convolution tensor and those of the vectors.

1.3 Tensorization by Means of Löwner Matrix (Löwner Folding)

A Löwner matrix of a vector $\mathbf{v} \in \mathbb{R}^{I+J}$ is formed from a function $f(t)$ sampled at $(I+J)$ distinct points $\{x_1, \ldots, x_I, y_1, \ldots, y_J\}$, to give

$$\mathbf{v} = [f(x_1), \ldots, f(x_I), f(y_1), \ldots, f(y_J)]^{\mathrm{T}} \in \mathbb{R}^{I+J},$$

so that the entries of $\mathbf{v}$ are partitioned into two disjoint sets, $\{f(x_i)\}_{i=1}^{I}$ and $\{f(y_j)\}_{j=1}^{J}$. The vector $\mathbf{v}$ is then converted into the Löwner matrix, $\mathbf{L} \in \mathbb{R}^{I \times J}$, defined by

$$\mathbf{L} = \left[\frac{f(x_i) - f(y_j)}{x_i - y_j} \right]_{ij} \in \mathbb{R}^{I \times J}.$$

Löwner matrices appear as a powerful tool in fitting a model to data in the form of a rational (Pade form) approximation, that is $f(x) = A(x)/B(x)$. When considered as transfer functions, such type of approximations are much more powerful than the polynomial approximations, as in this way it is also possible to model discontinuities and spiky data. The optimal order of such a rational approximation is given by the rank of the Löwner matrix. In the context of tensors, this allows us to construct a model of the original dataset which is amenable to higher-order tensor representation, has minimal computational complexity, and for which the accuracy is governed by the rank of the Löwner matrix. An example of Löwner folding of a vector $[1/3, 1/4, 1/5, 1/6, 1/8, 1/9, 1/10]$ is given below

$$\begin{bmatrix} \frac{1/3-1/8}{3-8} & \frac{1/3-1/9}{3-9} & \frac{1/3-1/10}{3-10} \\ \frac{1/4-1/8}{4-8} & \frac{1/4-1/9}{4-9} & \frac{1/4-1/10}{4-10} \\ \frac{1/5-1/8}{5-8} & \frac{1/5-1/9}{5-9} & \frac{1/5-1/10}{5-10} \\ \frac{1/6-1/8}{6-8} & \frac{1/6-1/9}{6-9} & \frac{1/6-1/10}{6-10} \end{bmatrix} = - \begin{bmatrix} 1/3 \\ 1/4 \\ 1/5 \\ 1/6 \end{bmatrix} \begin{bmatrix} 1/8 & 1/9 & 1/10 \end{bmatrix}.$$

More applications of this tensorization can be found in (Debals *et al.*, 2016a).

1.4 Tensorization based on Cumulant and Derivatives of the Generalised Characteristic Functions

The use of higher-order statistics (cumulants) or partial derivatives of the Generalised Characteristic Functions (GCF) as a means of tensorization is useful in the identification of a mixing matrix in a blind source separation.

Consider linear mixtures of R stationary sources, $\mathbf{S}$, received by an array of I sensors in the presence of additive noise, $\mathbf{N}$ (see Figure 1.4 for a general principle). The task is to estimate a mixing matrix $\mathbf{H} \in \mathbb{R}^{I \times R}$ from only the knowledge of the noisy observations

$$\mathbf{X} = \mathbf{HS} + \mathbf{N}, \tag{1.24}$$

under some mild assumptions, i.e., the sources are statistically independent and non-Gaussian, their number is known, and the matrix $\mathbf{H}$

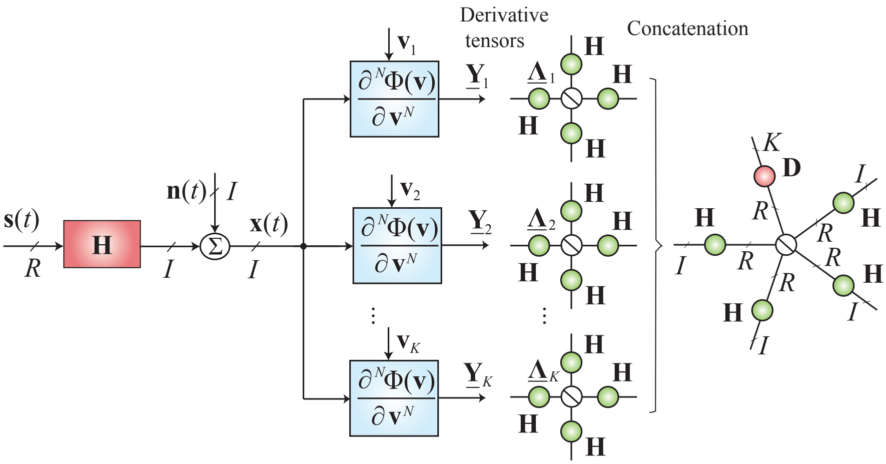


Figure 1.4: Tensorization based on derivatives of the characteristic functions and tensor-based approach to blind identification. The task is to estimate the mixing matrix, $\mathbf{H}$, from only the knowledge of the noisy output observations $\mathbf{X} = [\mathbf{x}(1), \ldots, \mathbf{x}(t), \ldots, \mathbf{x}(T)] \in \mathbb{R}^{I \times T}$, with $I < T$. A high dimensional tensor $\underline{\mathbf{Y}}$ is generated from the observations $\mathbf{X}$ by means of higher-order statistics (cumulants) or partial derivatives of the second generalised characteristic functions of the observations. A CP decomposition of $\underline{\mathbf{Y}}$ allows us to retrieve the mixing matrix $\mathbf{H}$.

has no pair-wise collinear columns (see also (Yeredor, 2000; Comon and Rajih, 2006))

A well-known approach to this problem is based on the decomposition of a high dimensional structured tensor, $\underline{\mathbf{Y}}$, generated from the observations, $\mathbf{X}$, by means of partial derivatives of the second GCFs of the observations at multiple processing points.

Derivatives of the GCFs. More specifically, we next show how to generate the tensor $\underline{\mathbf{Y}}$ from the observation, $\mathbf{X}$. We shall denote the first and second GCFs of the observations evaluated at a vector $\mathbf{u}$ of length I, respectively by

$$\phi_{\mathbf{x}}(\mathbf{u}) = \mathrm{E}\left[\exp(\mathbf{u}^{\mathrm{T}}\mathbf{x})\right], \quad \Phi_{\mathbf{x}}(\mathbf{u}) = \log \phi_{\mathbf{x}}(\mathbf{u})\,. \tag{1.25}$$

Similarly, $\phi_{\mathbf{s}}(\mathbf{v})$ and $\Phi_{\mathbf{s}}(\mathbf{v})$ designate the first and second GCFs of the sources, where $\mathbf{v}$ is of length R. Because the sources are statistically independent, the following holds

$$\Phi_{\mathbf{s}}(\mathbf{v}) = \Phi_{s_1}(v_1) + \Phi_{s_2}(v_2) + \cdots + \Phi_{s_R}(v_R)\,, \tag{1.26}$$

which implies that Nth-order derivatives of $\Phi_{\mathbf{s}}(\mathbf{v})$ with respect to $\mathbf{v}$ result in Nth-order diagonal tensors of size $R \times R \times \cdots \times R$, where $N = 2, 3, \ldots$, that is

$$\underline{\boldsymbol{\Psi}}_{\mathbf{s}}(\mathbf{v}) = \frac{\partial^N \Phi_{\mathbf{s}}(\mathbf{v})}{\partial \mathbf{v}^N} = \operatorname{diag}_N\left\{\frac{d^N\Phi_{s_1}}{dv_1^N}, \frac{d^N\Phi_{s_2}}{dv_2^N}, \ldots, \frac{d^N\Phi_{s_R}}{dv_R^N}\right\}. \tag{1.27}$$

In addition, for the noiseless case $\mathbf{x}(t) = \mathbf{H}\,\mathbf{s}(t)$, and since $\Phi_{\mathbf{x}}(\mathbf{u}) = \Phi_{\mathbf{s}}(\mathbf{H}^{\mathrm{T}}\mathbf{u})$, the Nth-order derivative of $\Phi_{\mathbf{x}}(\mathbf{u})$ with respect to $\mathbf{u}$ yields a symmetric tensor of Nth-order which admits a CP decomposition of rank-R with N identical factor matrices $\mathbf{H}$, to give

$$\underline{\boldsymbol{\Psi}}_{\mathbf{x}}(\mathbf{u}) = \underline{\boldsymbol{\Psi}}_{\mathbf{s}}(\mathbf{H}^{\mathrm{T}}\mathbf{u}) \times_1 \mathbf{H} \times_2 \mathbf{H} \cdots \times_N \mathbf{H}\,. \tag{1.28}$$

In order to improve the identification accuracy, the mixing matrix $\mathbf{H}$ should be estimated as a joint factor matrix in decompositions of various derivative tensors, evaluated at distinct processing points $\mathbf{u}_1$, $\mathbf{u}_2$, $\ldots$, $\mathbf{u}_K$. This is equivalent to a decomposition of an $(N+1)$th-order tensor $\underline{\mathbf{Y}}$ of size $I \times I \times \cdots \times I \times K$ concatenated from the K derivative tensors as

$$\underline{\mathbf{Y}}(:, \ldots, :, k) = \underline{\boldsymbol{\Psi}}_{\mathbf{x}}(\mathbf{u}_k), \quad k = 1, 2, \ldots, K. \tag{1.29}$$

The CP decomposition of the tensor $\underline{\mathbf{Y}}$ can be written in form of

$$\underline{\mathbf{Y}} = \underline{\mathbf{I}} \times_1 \mathbf{H} \times_2 \mathbf{H} \cdots \times_N \mathbf{H} \times_{N+1} \mathbf{D}\,, \tag{1.30}$$

where the last factor matrix $\mathbf{D}$ is of size $K \times R$, and each row comprises the diagonal of the symmetric tensor $\underline{\boldsymbol{\Psi}}_{\mathbf{s}}(\mathbf{H}^{\mathrm{T}}\mathbf{u}_k)$.

In the presence of statistically independent, additive and stationary Gaussian noise, we can eliminate the derivatives of the noise terms in the derivative tensor $\underline{\boldsymbol{\Psi}}_{\mathbf{x}}(\mathbf{u})$ by subtracting any other derivative tensor $\underline{\boldsymbol{\Psi}}_{\mathbf{x}}(\tilde{\mathbf{u}})$, or by an average of derivative tensors.

Estimation of Derivatives of GCF. In practice, the GCF of the observation and its derivatives are unknown, but can be estimated from the sample first GCF (Yeredor, 2000). Detailed expression and the approximation of the derivative tensor $\underline{\boldsymbol{\Psi}}_{\mathbf{x}}(\mathbf{u})$ for some low orders $N = 2, 3, \ldots, 7$, are given in Appendix 5.

Cumulants. When the derivative is taken at the origin, $\mathbf{u} = [0, \ldots, 0]^{\mathrm{T}}$, the tensor $\mathcal{K}_{\mathbf{x}}^{(N)} = \underline{\boldsymbol{\Psi}}_{\mathbf{x}}^{(N)}(\mathbf{0})$ is known as the Nth-order cumulant of $\mathbf{x}$, and a joint diagonalization or the CP decomposition of higher-order cumulants is a well-studied method for the estimation of the mixing matrix $\mathbf{H}$.

For the sources with symmetric probabilistic distributions, their odd-order cumulants, $N = 3, 5, \ldots$, are zero, and the cumulants of the mixtures are only due to noise. Hence, a decomposition of such tensors is not able to retrieve the mixing matrix. However, the odd-order cumulant tensors can be used to subtract the noise term in the derivative tensors evaluated at other processing points.

Example 1.6. Blind identification (BI) in a system of 2 mixtures and R binary signals.

To illustrate the efficiency of higher-order derivatives of the second GCF in blind identification we considered a system of two mixtures, $I = 2$, linearly composed by R signals of length $T = 100 \times 2^R$, the entries of which can take the values 1 or -1, i.e., $s_{r,t} = 1$ or -1. The mixing matrix $\mathbf{H}$ of size $2 \times R$ was randomly generated, where $R = 4, 6, 8$. The signal-to-noise ratio was SNR = 20 dB. The main purpose of BI is to estimate the mixing matrix $\mathbf{H}$.

We constructed 50 tensors $\underline{\mathbf{Y}}_i$ $(i = 1, \ldots, 50)$ of size $R \times \cdots \times R \times 3$, which comprise three derivative tensors evaluated at the two leading left singular vectors of $\mathbf{X}$, and a unit-length processing point, generated such that its collinearity degree with the first singular vector uniformly distributed over a range of $[-0.99, 0.99]$. The average derivative tensor was used to eliminate the noise term in $\underline{\mathbf{Y}}_i$.

CP decomposition of derivative tensors. The tensors $\underline{\mathbf{Y}}_i$ were decomposed by CP decompositions of rank-R to retrieve the mixing matrix $\mathbf{H}$. The mean of Squared Angular Errors $SAE(\mathbf{h}_r, \hat{\mathbf{h}}_r) = -20 \log_{10} \arccos(\frac{\mathbf{h}_r^{\mathrm{T}} \hat{\mathbf{h}}_r}{|\mathbf{h}_r|_2 |\hat{\mathbf{h}}_r|_2})$ over all columns $\mathbf{h}_r$ was computed as a performance index for one estimation of the mixing matrix.

The averages of the mean and best MSAEs over 100 independent runs for the number of the unknown sources $R = 4, 6, 8$ are plotted in

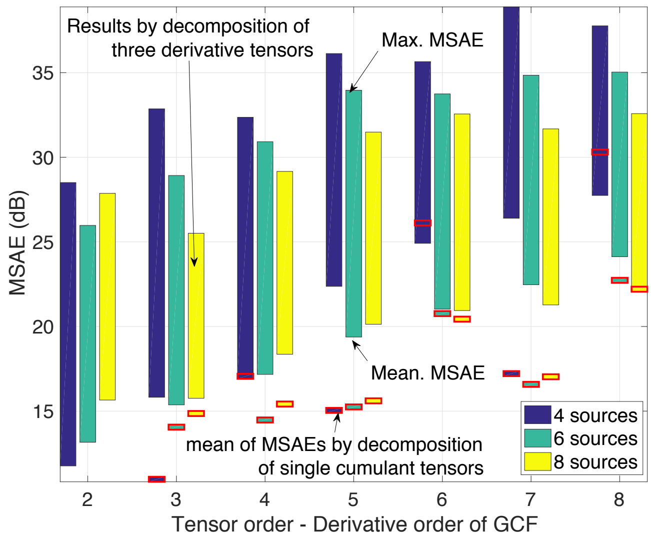


Figure 1.5: Mean SAE (in dB) in the estimation of the mixing matrix $\mathbf{H}$ from only two mixtures, achieved by CP decomposition of three $2 \times 2 \times \cdots \times 2$ derivative tensors of the second GCFs. Small bars in red represent the mean of MSAEs, obtained by decomposition of single cumulant tensors.

Figure 1.5. The results indicate that with a suitably chosen processing point, the decomposition of the derivative tensors yielded good estimation of the mixing matrix. Of more importance is that higher-order derivative tensors, e.g., 7th and 8th orders, yielded better performance than lower-order tensors, while the estimation accuracy deteriorated with the number of sources.

CP decomposition of cumulant tensors. Because of symmetric pdfs, the odd order cumulants of the sources are zero. Only decompositions of cumulants of order 6 or 8 were able to retrieve the mixing matrix $\mathbf{H}$. For all the test cases, better performances could be obtained by a decomposition of three derivative tensors.

Tensor train decomposition of derivative tensors. The estimation of the mixing matrix $\mathbf{H}$ can be performed in a two-stage decomposition

- A tensor train decomposition of high-order derivative tensors, e.g., tensor order exceeds 5.

- A CP decomposition of the tensor in TT-format, to retrieve the mixing matrix.

Experimental results confirmed that the performances with prior TT-decomposition were more stable and yielded an approximately 2 dB higher mean SAE than those using only CP decomposition for derivative tensors of orders 7 and 8 and a relatively high number of unknown sources.

1.4.1 Tensor Structures in Constant Modulus Signal Separation

Another method to generate tensors of relatively high order in BSS is through modelling modulus of the estimated signals as roots of a polynomial.

Consider a linear mixing system $\mathbf{X} = \mathbf{HS}$ with R sources of length K, and I mixtures, where the modulus of the sources $\mathbf{S}$ is drawn from a set of given moduli. For simplicity, we assume $I = R$. For example, the binary phase-shift keying (BPSK) signal in telecommunication consists of a sequence of 1 and -1, hence, it has a constant modulus of unity. The quadrature phase shift keying (QPSK) signal takes one of the values $\pm 1 \pm 1i$, i.e., it has a constant modulus $\sqrt{2}$. The 16-QAM signal has three squared moduli of 2, 10 and 18. For this BSS problem for single constant modulus signals, Lathauwer (2004) linked the problem to CP decomposition of a fourth-order tensor. For multi-constant modulus signals, Debals *et al.* (2016b) established a link to a coupled CP decomposition.

A common method to extract the original sources $\mathbf{S}$ is to use a demixing matrix $\mathbf{W}$ of size $I \times R$ or a vector $\mathbf{w}$ of length I such that $\mathbf{y} = \mathbf{w}^{\mathrm{T}}\mathbf{X}$ is an estimate of one of the source signals. The constant modulus constraints require that each entry, $|y_k|$, must be one of given moduli, $c_1, c_2, \ldots, c_M$. This means that for all entries of $\mathbf{y}$ the following holds

$$f(y_k) = \prod_{m=1}^{M} (|y_k|^2 - c_m) = 0\,. \tag{1.31}$$

In other words, $|y_k|^2$ are roots of an Mth-degree polynomial, given by

$$p^M + \alpha_m p^{M-1} + \cdots + \alpha_2 p + \alpha_1\,,$$

with coefficients $\alpha_{M+1} = 1$, and $\alpha_1, \alpha_2, \ldots, \alpha_M$, given by

$$\alpha_m = (-1)^{m-1} \sum_{i_1, i_2, \ldots, i_m} c_{i_1} c_{i_2} \cdots c_{i_m} \,. \tag{1.32}$$

By expressing $|y_k|^2 = (\mathbf{w} \otimes \mathbf{w}^*)^{\mathrm{T}} (\mathbf{x}_k \otimes \mathbf{x}_k^*)$, and

$$|y_k|^{2m} = (\mathbf{w}^{\otimes m} \otimes (\mathbf{w}^{\otimes m})^*)^{\mathrm{T}} (\mathbf{x}_k^{\otimes m} \otimes (\mathbf{x}_k^{\otimes m})^*) \,,$$

where the symbol "$*$" represents the complex conjugate, $\mathbf{x}^{\otimes m} = \mathbf{x} \otimes \mathbf{x} \otimes \cdots \otimes \mathbf{x}$ denotes the Kronecker product of m vectors $\mathbf{x}$, and bearing in mind that the rank-1 tensors $\mathbf{w}^{\circ m} = \mathbf{w} \circ \mathbf{w} \circ \cdots \circ \mathbf{w}$ are symmetric, and in general have only $\frac{(R+m-1)!}{m!(R-1)!}$ distinct coefficients, the rank-1 tensors $\mathbf{w}^{\circ m} \circ (\mathbf{w}^{\circ m})^*$ have at least $\left(\frac{(R+m-1)!}{m!(R-1)!}\right)^2$ distinct entries. We next introduce the operator $\mathcal{K}$ which keeps only distinct entries of the symmetric tensor $\mathbf{w}^{\circ m} \circ (\mathbf{w}^{\circ m})^*$ or of the vector $\mathbf{w}^{\otimes m} \otimes (\mathbf{w}^{\otimes m})^*$. The constant modulus constraint of y_k can then be rewritten as

$$\begin{aligned} f(y_k) &= \alpha_1 + \sum_{m=2}^{M+1} \alpha_m (\mathbf{w}^{\otimes m} \otimes (\mathbf{w}^{\otimes m})^*)^{\mathrm{T}} (\mathbf{x}_k^{\otimes m} \otimes (\mathbf{x}_k^{\otimes m})^*) \\ &= \alpha_1 + \sum_{m=2}^{M+1} \alpha_m (\mathcal{K}(\mathbf{w}^{\otimes m} \otimes (\mathbf{w}^{\otimes m})^*))^{\mathrm{T}} \operatorname{diag}(\mathbf{d}_m)\, \mathcal{K}(\mathbf{x}_k^{\otimes m} \otimes (\mathbf{x}_k^{\otimes m})^*) \\ &= \alpha_1 + \left[\ldots, (\mathcal{K}(\mathbf{w}^{\otimes m} \otimes (\mathbf{w}^{\otimes m})^*))^{\mathrm{T}}, \ldots\right] \\ &\qquad [\ldots, \mathcal{K}(\mathbf{x}_k^{\otimes m} \otimes (\mathbf{x}_k^{\otimes m})^*)^{\mathrm{T}} \operatorname{diag}(\alpha_m \mathbf{d}_m), \ldots]^{\mathrm{T}} \,, \end{aligned}$$

where $d_m(i)$ represents the number of occurrences of an entry of $\mathcal{K}(\mathbf{x}_k^{\otimes m} \otimes (\mathbf{x}_k^{\otimes m})^*)$ in $\mathbf{x}_k^{\otimes m} \otimes (\mathbf{x}_k^{\otimes m})^*$.

The vector of the constant modulus constraints of $\mathbf{y}$ is now given by

$$\boldsymbol{f} = [\ldots, f(y_k), \ldots]^{\mathrm{T}} = \alpha_1 \mathbf{1} + \mathbf{Q}\mathbf{v} \,, \tag{1.33}$$

where

$$\mathbf{v} = \begin{bmatrix} \vdots \\ \mathcal{K}(\mathbf{w}^{\otimes m} \otimes (\mathbf{w}^{\otimes m})^*) \\ \vdots \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \vdots \\ \operatorname{diag}(\alpha_m \mathbf{d}_m) \mathcal{K}(\mathbf{X}^{\odot m} \odot (\mathbf{X}^{\odot m})^*) \\ \vdots \end{bmatrix}^{\mathrm{T}} .$$

The constraint vector is zero for the exact case, and should be small for the noisy case. For the exact case, from (1.33) and $f(y_{k+1}) - f(y_k) = 0$, this leads to

$$\mathbf{L}\mathbf{Q}\mathbf{v} = \mathbf{0}\,,$$

where $\mathbf{L}$ is the first-order Laplacian implying that the vector $\mathbf{v}$ is in the null space of the matrix $\tilde{\mathbf{Q}} = \mathbf{L}\mathbf{Q}$. The above condition holds for other demixing vectors $\mathbf{w}$, i.e., $\tilde{\mathbf{Q}}\mathbf{V} = \mathbf{0}$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_R]$, and each $\mathbf{v}_r$ is constructed from a corresponding demixing vector $\mathbf{w}_r$.

With the assumption $I = R$, and that the sources have complex values, and the mixing matrix does not have collinear columns, it can be shown that the kernel of the matrix $\tilde{\mathbf{Q}}$ has the dimension of R (Debals *et al.*, 2016b). Therefore, the basis vectors, $\mathbf{z}_r$, $r = 1, \ldots, R$, of the kernel of $\tilde{\mathbf{Q}}$ can be represented as linear combination of $\mathbf{V}$, that is

$$\mathbf{z}_r = \mathbf{V}\boldsymbol{\lambda}_r\,.$$

Next we partition $\mathbf{z}_r$ into M parts, $\mathbf{z}_r = [\mathbf{z}_{rm}]$, each of the length $\left(\frac{(R+m-1)!}{m!(R-1)!}\right)^2$, which can be expressed as

$$\mathbf{z}_{rm} = \sum_{s=1}^{R} \lambda_{rs}\mathcal{K}(\mathbf{w}_s^{\otimes m} \otimes (\mathbf{w}_s^{\otimes m})^*) = \mathcal{K}\left(\sum_{s=1}^{R} \lambda_{rs}\mathbf{w}_s^{\otimes m} \otimes (\mathbf{w}_s^{\otimes m})^*\right),$$

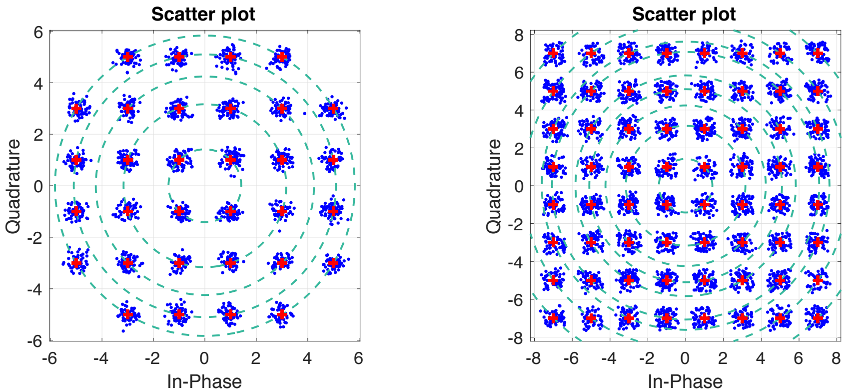
thus implying that $\mathbf{W}$ and $\mathbf{W}^*$ are factor matrices of a symmetric tensor $\underline{\mathbf{Z}}_{rm}$ of $(2m)$th-order, constructed from the vector $\mathbf{z}_{rm}$, i.e., $\mathcal{K}(\mathrm{vec}(\underline{\mathbf{Z}}_{rm})) = \mathbf{z}_{rm}$, in the form

$$\underline{\mathbf{Z}}_{rm} = [\![\mathrm{diag}_{2m}(\boldsymbol{\lambda}_r); \underbrace{\mathbf{W}, \ldots, \mathbf{W}}_{m \text{ terms}}, \underbrace{\mathbf{W}^*, \ldots, \mathbf{W}^*}_{m \text{ terms}}]\!]\,. \tag{1.34}$$

By concatenating all R tensors $\underline{\mathbf{Z}}_{1m}, \ldots, \underline{\mathbf{Z}}_{Rm}$ into one $(2m+1)$th-order tensor $\underline{\mathbf{Z}}_m$, the above R CP decompositions become

$$\underline{\mathbf{Z}}_m = [\![\underline{\mathbf{I}}; \underbrace{\mathbf{W}, \ldots, \mathbf{W}}_{m \text{ terms}}, \underbrace{\mathbf{W}^*, \ldots, \mathbf{W}^*}_{m \text{ terms}}, \boldsymbol{\Lambda}]\!]\,. \tag{1.35}$$

All together, the M CP decompositions of $\underline{\mathbf{Z}}_1, \ldots, \underline{\mathbf{Z}}_M$ form a coupled CP tensor decomposition to find the two matrices $\mathbf{W}$ and $\boldsymbol{\Lambda}$.



(a) 32-QAM (b) 64-QAM

Figure 1.6: Scatter plots of the estimated sources (blue dots). Red dots indicate the ideal signal constellation, the values of which are located on one of dashed circles.

Example 1.7 (Separation of QAM signals.). We performed the separation of two rectangular 32- or 64-QAM signals of length 1000 from two mixture signals corrupted by additive Gaussian noise with SNR = 15 dB. Columns of the real-valued mixing matrix had unit-length, and a pair-wise collinearity of 0.4. The 32-QAM signal had $M = 5$ constant moduli of 2, 10, 18, 26 and 34, whereas the 64-QAM signal had $M = 9$ squared constant moduli of 2, 10, 18, 26, 34, 50, 58, 74 and 98. Therefore, for the first case (32-QAM), the demixing matrix was estimated from 5 tensors of size $2 \times 2 \times \cdots \times 2$ and of respective orders 3, 5, 7, 9 and 11, while for the later case (64-QAM), we decomposed 9 quantized tensors of orders $3, 5, \ldots, 19$. The estimated QAM signals for the two cases were perfectly reconstructed with zero bit error rates. Scatter plots of the recovered signals are shown in Figure 1.6.

1.5 Tensorization by Learning Local Structures

Different from the previous tensorizations, this tensorization approach generates tensors from local blocks (patches) which are similar or closely related. For the example of an image, given that the intensities of pixels in a small window are highly correlated, hidden structures

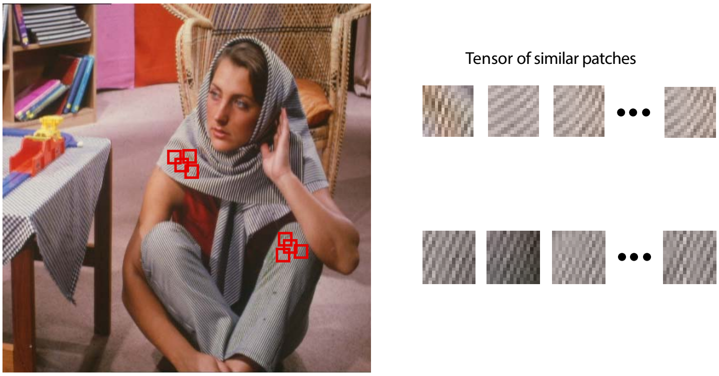


Figure 1.7: A "local-structure" tensorization method generates 5th-order tensors of size $h \times w \times 3 \times (2d+1) \times (2d+1)$ from similar image patches, or patches in close spatial proximity.

which represent relations between small patches of pixels can be learnt in local areas. These structures can then be used to reconstruct the image as a whole in, e.g., an application of image denoising (Phan *et al.*, 2016).

For a color RGB image $\underline{\mathbf{Y}}$ of size $I \times J \times 3$, each block of pixels of size $h \times w \times 3$ is denoted as

$$\underline{\mathbf{Y}}_{r,c} = \underline{\mathbf{Y}}(r : r+h-1, c : c+w-1, :).$$

A small tensor, $\underline{\mathbf{Z}}_{r,c}$, of size $h \times w \times 3 \times (2d+1) \times (2d+1)$, comprising $(2d+1)^2$ blocks centered around $\underline{\mathbf{Y}}_{r,c}$, with d denoting the neighbourhood width, can be constructed in the form

$$\underline{\mathbf{Z}}_{r,c}(:, :, :, d+1+i, d+1+j) = \underline{\mathbf{Y}}_{r+i,c+j},$$

where $i, j = -d, \ldots, 0, \ldots, d$, as illustrated in Figure 1.7. Every (r, c)-th block $\underline{\mathbf{Z}}_{r,c}$ is then approximated through a constrained tensor decomposition

$$\|\underline{\mathbf{Z}}_{r,c} - \hat{\underline{\mathbf{Z}}}_{r,c}\|_F^2 \leqslant \varepsilon^2 \,, \tag{1.36}$$

where the noise level ε^2 can be determined by inspecting the coefficients of the image in the high-frequency bands. A pixel is then reconstructed as the average of all its approximations which cover that pixel.

(a) Noisy image (b) TT, PSNR = 31.64 dB (c) CP, PSNR = 28.90 dB

Figure 1.8: Tensor based image reconstruction in Example 1.8. The Pepper image with added noise at 10 dB SNR (left), and the images reconstructed using the TT (middle) and CP (right) decompositions.

Example 1.8. Image denoising. The principle of tensorization from learning the local structures is next demonstrated in an image denoising application for the benchmark "peppers" color image of size $256 \times 256 \times 3$, which was corrupted by white Gaussian noise at SNR = 10 dB. Latent structures were learnt for patches of sizes $8 \times 8 \times 3$ (i.e., $h = w = 8$) in the search area of width $d = 3$. To the noisy image, we applied the DCT spatial filtering before their block reconstruction. The results are shown in Figure 1.8, and illustrate the advantage of the tensor network approach over a CP decomposition approach.

1.6 Tensorization based on Divergences, Similarities or Information Exchange

For a set of I data points $\mathbf{x}_i$, $i = 1, 2, \ldots, I$, this type of tensorization generates an Nth-order nonnegative symmetric tensor of size $I \times I \times \cdots \times I$, the entries of which represent N-way similarities or dissimilarities between $\mathbf{x}_{i_1}$, $\mathbf{x}_{i_2}$, ..., $\mathbf{x}_{i_N}$, where $i_n = 1, \ldots, I$, so that

$$\underline{\mathbf{Y}}(i_1, i_2, \ldots, i_N) = d(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_N})\,. \tag{1.37}$$

Such metric function can express pair-wise distances between the two observations $\mathbf{x}_i$ and $\mathbf{x}_j$. In a general case, $d(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_N})$ can compute the volume of a convex hull formed by N data points.

The so generated tensor can be expanded to $(N+1)$th-order tensor, where the last mode expresses the change of data points over e.g., time or trials. Tensorizations based on divergences and similarities are useful for the analysis of interaction between observed entities, and for their clustering or classification.

1.7 Tensor Structures in Multivariate Polynomial Regression

The Multivariate Polynomial Regression (MPR) is an extension of the linear and multilinear regressions which allows us to model nonlinear interaction between independent variables (Chen and Billings, 1989; Billings, 2013; Vaccari, 2003). For illustration, consider a simple example of fitting a curve to data with two independent variables x_1 and x_2, in the form

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_{12} x_1 x_2 \,. \tag{1.38}$$

The term w_{12} then quantifies the strength of interaction between the two independent variables in the data, x_1 and x_2. Observe that the model is still linear with respect to the variables x_1 and x_2, while involving the cross-term $w_{12}x_1x_2$. The above model can also have more terms, e.g., $x_1^2, x_1x_2^2$, to describe more complex functional behaviours. For example, the full quadratic polynomial regression for two independent variables, x_1 and x_2, can have up to 9 terms, given by

$$\begin{aligned} y = {} & w_0 + w_1 x_1 + w_2 x_2 + w_{12} x_1 x_2 \\ & + w_{11} x_1^2 + w_{22} x_2^2 + w_{112} x_1^2 x_2 + w_{122} x_1 x_2^2 + w_{1122} x_1^2 x_2^2 \,. \end{aligned} \tag{1.39}$$

Tensor representation of the system weights. The simple model for two independent variables in (1.38) can be rewritten in a bilinear form as

$$y = \begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} w_0 & w_2 \\ w_1 & w_{12} \end{bmatrix} \begin{bmatrix} 1 \\ x_2 \end{bmatrix},$$

whereas the full model in (1.39) has an equivalent bilinear expression

$$y = \begin{bmatrix} 1 & x_1 & x_1^2 \end{bmatrix} \begin{bmatrix} w_0 & w_2 & w_{22} \\ w_1 & w_{12} & w_{122} \\ w_{11} & w_{112} & w_{1122} \end{bmatrix} \begin{bmatrix} 1 \\ x_2 \\ x_2^2 \end{bmatrix},$$

or a tensor-vector product representation

$$y = \underline{\mathbf{W}} \,\bar{\times}_1 \begin{bmatrix} 1 \\ x_1 \end{bmatrix} \bar{\times}_2 \begin{bmatrix} 1 \\ x_1 \end{bmatrix} \bar{\times}_3 \begin{bmatrix} 1 \\ x_2 \end{bmatrix} \bar{\times}_4 \begin{bmatrix} 1 \\ x_2 \end{bmatrix}, \qquad (1.40)$$

where the 4th-order weight tensor $\underline{\mathbf{W}}$ is of size $2 \times 2 \times 2 \times 2$, and is given by

$$\underline{\mathbf{W}}(:,:,1,1) = \begin{bmatrix} w_0 & \frac{1}{2}w_1 \\ \frac{1}{2}w_1 & w_{11} \end{bmatrix}, \ \underline{\mathbf{W}}(:,:,2,2) = \begin{bmatrix} w_{22} & \frac{1}{2}w_{122} \\ \frac{1}{2}w_{122} & w_{1122} \end{bmatrix},$$
$$\underline{\mathbf{W}}(:,:,1,2) = \underline{\mathbf{W}}(:,:,2,1) = \frac{1}{2}\begin{bmatrix} w_2 & \frac{1}{2}w_{12} \\ \frac{1}{2}w_{12} & w_{112} \end{bmatrix}.$$

It is now obvious that for a generalised system with N independent variables, $x_1, \ldots, x_N$, the MPR can be written as a tensor-vector product as (Chen and Billings, 1989)

$$\begin{aligned} y &= \sum_{i_1=0}^{N} \sum_{i_2=0}^{N} \cdots \sum_{i_N=0}^{N} w_{i_1,i_2,\ldots,i_N}\, x_1^{i_1}\, x_2^{i_2} \cdots x_N^{i_N} \\ &= \underline{\mathbf{W}} \,\bar{\times}_1\, \mathcal{V}_N(x_1) \,\bar{\times}_2\, \mathcal{V}_N(x_2) \cdots \bar{\times}_N\, \mathcal{V}_N(x_N)\,, \end{aligned} \qquad (1.41)$$

where $\underline{\mathbf{W}}$ is an Nth-order tensor of size $(N+1)\times(N+1)\times\cdots\times(N+1)$, and $\mathcal{V}_N(x)$ is the length-$(N+1)$ Vandermonde vector of x, given by

$$\mathcal{V}_N(x) = \begin{bmatrix} 1 & x & x^2 & \ldots & x^N \end{bmatrix}^{\mathrm{T}}. \qquad (1.42)$$

Similarly to the representation in (1.40), the MPR model in (1.41) can be equivalently expressed as a product of a tensor of N^2th-order and size $2 \times 2 \times \cdots \times 2$ with N vectors of length-2, to give

$$y = \underline{\widetilde{\mathbf{W}}} \,\bar{\times}_{1:N} \begin{bmatrix} 1 \\ x_1 \end{bmatrix} \bar{\times}_{N+1:2N} \begin{bmatrix} 1 \\ x_2 \end{bmatrix} \cdots \bar{\times}_{N(N-1)+1:N^2} \begin{bmatrix} 1 \\ x_N \end{bmatrix} (1.43)$$

An illustration of the MPR is given in Figure 1.9, where the input units are scalars.

The MPR has found numerous applications, owing to its ability to model any smooth, continuous nonlinear input-output system , see e.g. (Vaccari, 2003). However, since the number of parameters in the model in (1.41) grows exponentially with the number of variables, N,

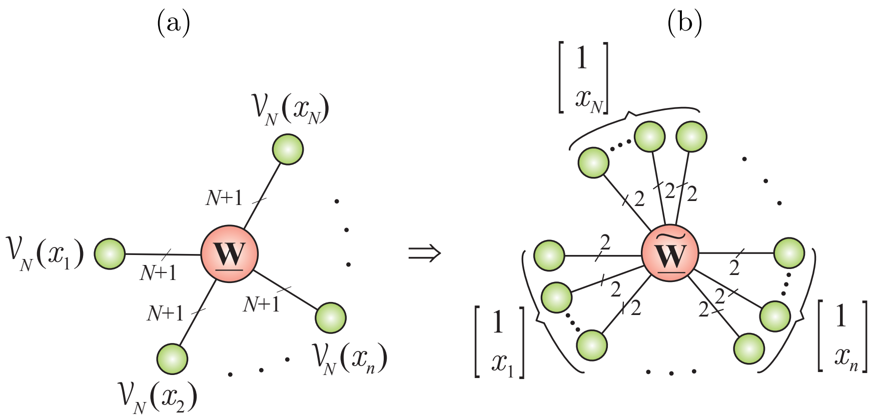


Figure 1.9: Graphical illustration of Multivariate Polynomial Regression (MPR). (a) The MPR for multiple input units $x_1, \ldots, x_N$, where the nonlinear function $h(x_1, \ldots, x_N)$ is expressed as a multilinear tensor-vector product of an Nth-order tensor, $\underline{\mathbf{W}}$, of size $(N+1)\times(N+1)\times\cdots\times(N+1)$, and Vandermonde vectors $\mathcal{V}_N(x_n)$ of length $(N+1)$. (b) An equivalent MPR model but with quantized N^2th-order tensor $\widetilde{\underline{\mathbf{W}}}$ of size $2 \times 2 \times \cdots \times 2$.

the MPR demands a huge amount of data in order to yield a good model, and therefore, it is computationally intensive in a raw tensor format, and thus not suitable for very high-dimensional data. To this end, low-rank tensor network representation emerges as a viable approach to accomplishing MPR. For example, the weight tensor $\underline{\mathbf{W}}$ can be constrained to be in low rank TT-format (Chen *et al.*, 2016). An alternative approach would be to consider a truncated model which takes only two entries along each mode of $\underline{\mathbf{W}}$ in (1.41). In other words, this truncated model becomes linear with respect to each variable x_n (Novikov *et al.*, 2016), leading to

$$y = \underline{\mathbf{W}}_t \,\bar{\times}_1 \begin{bmatrix} 1 \\ x_1 \end{bmatrix} \bar{\times}_2 \begin{bmatrix} 1 \\ x_2 \end{bmatrix} \cdots \bar{\times}_N \begin{bmatrix} 1 \\ x_N \end{bmatrix}, \tag{1.44}$$

where $\underline{\mathbf{W}}_t$ is a tensor of size $2\times2\times\cdots\times2$ in the QTT-format. Both (1.43) and (1.44) represent the weight tensors in the QTT-format, however, the tensor $\widetilde{\underline{\mathbf{W}}}$ in (1.43) has N^2 core tensors of the full MPR, whereas $\underline{\mathbf{W}}_t$ in (1.44) has N core tensors for the truncated model.

1.8 Tensor Structures in Vector-variate Regression

The MPR in (1.41) is formulated for scalar data. When the observations are vectors or tensors, the model can be extended straightforwardly. For illustration, consider a simple case of two independent vector inputs $\mathbf{x}_1$ and $\mathbf{x}_2$. Then, the nonlinear function which maps the input to the output $y = h(\mathbf{x}_1, \mathbf{x}_2)$ can be approximated in a linear form as

$$\begin{aligned} y = h(\mathbf{x}_1, \mathbf{x}_2) &= w_0 + \mathbf{w}_1^{\mathrm{T}}\mathbf{x}_1 + \mathbf{w}_2^{\mathrm{T}}\mathbf{x}_2 + \mathbf{x}_1^{\mathrm{T}}\mathbf{W}_{12}\mathbf{x}_2 \\ &= [1,\ \mathbf{x}_1^{\mathrm{T}}] \begin{bmatrix} w_0 & \mathbf{w}_2^{\mathrm{T}} \\ \mathbf{w}_1 & \mathbf{W}_{12} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x}_2 \end{bmatrix}, \end{aligned} \tag{1.45}$$

or in a quadratic with 9 terms, including one bias, two vectors, three matrices, two third-order tensors and one fourth-order tensor, given by

$$\begin{aligned} h(\mathbf{x}_1, \mathbf{x}_2) &= w_0 + \mathbf{w}_1^{\mathrm{T}}\mathbf{x}_1 + \mathbf{w}_2^{\mathrm{T}}\mathbf{x}_2 + \mathbf{x}_1^{\mathrm{T}}\mathbf{W}_{12}\mathbf{x}_2 + \mathbf{x}_1^{\mathrm{T}}\,\mathbf{W}_{11}\mathbf{x}_1 + \mathbf{x}_2^{\mathrm{T}}\,\mathbf{W}_{22}\mathbf{x}_2 \\ &\quad + \underline{\mathbf{W}}_{112}\,\bar{\times}_1\,\mathbf{x}_1\,\bar{\times}_2\,\mathbf{x}_1\,\bar{\times}_3\,\mathbf{x}_2 + \underline{\mathbf{W}}_{122}\,\bar{\times}_1\,\mathbf{x}_1\,\bar{\times}_2\,\mathbf{x}_2\,\bar{\times}_3\,\mathbf{x}_2 \\ &\quad + \underline{\mathbf{W}}_{1122}\,\bar{\times}_1\,\mathbf{x}_1\,\bar{\times}_2\,\mathbf{x}_1\,\bar{\times}_3\,\mathbf{x}_2\,\bar{\times}_4\,\mathbf{x}_2 \\ &= [1,\ \mathbf{x}_1^{\mathrm{T}},\ (\mathbf{x}_1 \otimes \mathbf{x}_1)^T]\ \mathbf{W} \begin{bmatrix} 1 \\ \mathbf{x}_2 \\ \mathbf{x}_2 \otimes \mathbf{x}_2 \end{bmatrix}, \end{aligned}$$

where the matrix $\mathbf{W}$ is given

$$\mathbf{W} = \begin{bmatrix} w_0 & \mathbf{w}_2^{\mathrm{T}} & \mathrm{vec}(\mathbf{W}_{22})^{\mathrm{T}} \\ \mathbf{w}_1 & \mathbf{W}_{12} & [\underline{\mathbf{W}}_{122}]_{(1)} \\ \mathrm{vec}(\mathbf{W}_{11}) & [\underline{\mathbf{W}}_{112}]_{(1,2)} & [\underline{\mathbf{W}}_{1122}]_{(1,2)} \end{bmatrix}. \tag{1.46}$$

and $[\underline{\mathbf{W}}_{112}]_{(1,2)}$ represents the mode-(1,2) unfolding of the tensor $\underline{\mathbf{W}}_{112}$. Similarly to (1.40), the above model has an equivalent expression of through the tensor-vector product of a fourth-order tensor $\underline{\mathbf{W}}$, in the form

$$y = \underline{\mathbf{W}}\,\bar{\times}_1 \begin{bmatrix} 1 \\ \mathbf{x}_1 \end{bmatrix} \bar{\times}_2 \begin{bmatrix} 1 \\ \mathbf{x}_1 \end{bmatrix} \bar{\times}_3 \begin{bmatrix} 1 \\ \mathbf{x}_2 \end{bmatrix} \bar{\times}_4 \begin{bmatrix} 1 \\ \mathbf{x}_2 \end{bmatrix}. \tag{1.47}$$

In general, the regression for a system with N input vectors, $\mathbf{x}_n$ of lengths I_n, can be written as

$$h(\mathbf{x}_1, \ldots, \mathbf{x}_N) = w_0 + \sum_{d=1}^{N^2} \sum_{i_1, i_2, \ldots, i_d=1}^{N} \underline{\mathbf{W}}_{i_1, i_2, \ldots, i_d} \bar{\times} \left(\mathbf{x}_{i_1} \circ \mathbf{x}_{i_2} \circ \cdots \circ \mathbf{x}_{i_d}\right), \tag{1.48}$$

where $\bar{\times}$ represents the inner product between two tensors, and the tensors $\underline{\mathbf{W}}_{i_1,\ldots,i_d}$ are of d-th order, and of size $I_{i_1} \times I_{i_2} \times \cdots \times I_{i_d}$, $d = 1, \ldots, N^2$. The representation of the generalised model as a tensor-vector product of an Nth-order tensor of size $J_1 \times J_2 \times \cdots \times J_N$, where $J_n = \frac{I_n^{N+1}-1}{I_n - 1}$, comprising all the weights, is given by

$$h(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \underline{\mathbf{W}} \,\bar{\times}_1\, \mathbf{v}_N(\mathbf{x}_1) \,\bar{\times}_2\, \mathbf{v}_N(\mathbf{x}_2) \cdots \bar{\times}_N\, \mathbf{v}_N(\mathbf{x}_N), \tag{1.49}$$

where

$$\mathbf{v}_N(\mathbf{x}) = \begin{bmatrix} 1 & \mathbf{x}^{\mathrm{T}} & (\mathbf{x} \otimes \mathbf{x})^T & \ldots & (\mathbf{x} \otimes \cdots \otimes \mathbf{x})^T \end{bmatrix}^{\mathrm{T}}, \tag{1.50}$$

or, in a more compact form, with a very high-order tensor $\widetilde{\underline{\mathbf{W}}}$ of N^2th-order and of size $(I_1 + 1) \times \cdots \times (I_1 + 1) \times (I_2 + 1) \times \cdots \times (I_N + 1) \times \cdots \times (I_N + 1)$, as

$$h(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \widetilde{\underline{\mathbf{W}}} \,\bar{\times}_{1:N} \begin{bmatrix} 1 \\ \mathbf{x}_1 \end{bmatrix} \cdots \bar{\times}_{N(N-1)+1:N^2} \begin{bmatrix} 1 \\ \mathbf{x}_N \end{bmatrix} \tag{1.51}$$

The illustration of this generalized model is given in Figure 1.10.

Tensor-variate model. When the observations are matrices, $\mathbf{X}_n$, or higher-order tensors, $\underline{\mathbf{X}}_n$, the models in (1.48), (1.49) and (1.51) are still applicable and operate by replacing the original vectors, $\mathbf{x}_n$, by the vectorization of the higher-order inputs. This is because the inner product between two tensors can be expressed as a product of their two vectorizations.

Separable representation of the weights. Similar to the MPR, the challenge in the generalised tensor-variate regression is the curse of dimensionality of the weight tensor $\underline{\mathbf{W}}$ in (1.49), or of the tensor $\widetilde{\underline{\mathbf{W}}}$ in (1.51).

A common method to deal with the problem is to restrict the model to some low order, i.e., to the first order. The weight tensor is now only

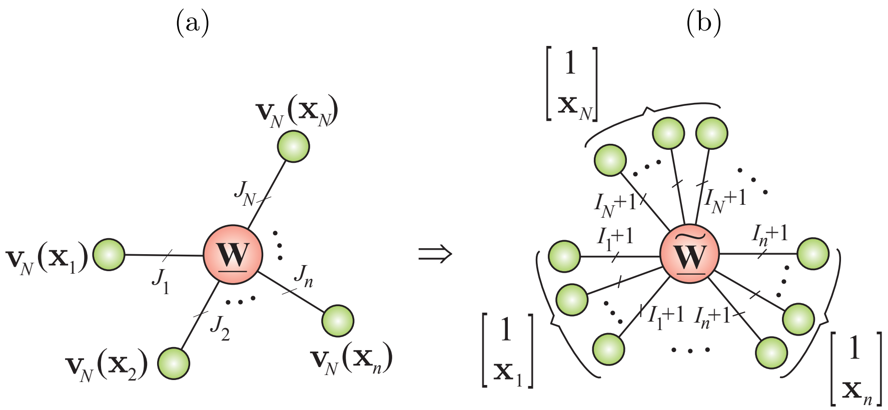


Figure 1.10: Graphical illustration of the vector-variate regression. (a) The vector-variate regression for multiple input units $\mathbf{x}_1, \ldots, \mathbf{x}_N$, where the nonlinear function $h(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ is expressed as a tensor-vector product of an Nth-order core tensor, $\underline{\mathbf{W}}$, of size $J_1 \times J_2 \times \cdots \times J_N$, and Vandermonde-like vectors $\mathbf{v}_N(\mathbf{x}_n)$ of length J_n, where $J_n = \frac{I_n^{N+1}-1}{I_n-1}$. (b) An equivalent regression model but with an N^2th-order tensor of size $(I_1+1) \times \cdots \times (I_1+1) \times (I_2+1) \times \cdots \times (I_N+1) \times \cdots \times (I_N+1)$. When the input units are scalars, the tensor $\widetilde{\underline{\mathbf{W}}}$ is of size $2 \times 2 \times \cdots \times 2$.

of size $(I_1 + 1) \times (I_2 + 1) \times \cdots \times (I_N + 1)$. The large weight tensor can then be represented in the canonical form (Nguyen *et al.*, 2015; Qi *et al.*, 2016), the TT/MPS tensor format (Stoudenmire and Schwab, 2016), or the hierarchical Tucker tensor format (Cohen and Shashua, 2016).

1.9 Tensor Structure in Volterra Models of Nonlinear Systems

1.9.1 Discrete Volterra Model

System identification is a paradigm which aims to provide a mathematical description of a system from the observed system inputs and outputs (Billings, 2013). In practice, tensors are inherently present in *Volterra operators* which model the system response of a nonlinear system which maps an input signal $x(t)$ to an output signal $y(t)$ in the form

$$y(t) = V(x(t)) = h_0 + H_1(x(t)) + H_2(x(t)) + \cdots + H_n(x(t)) + \cdots$$

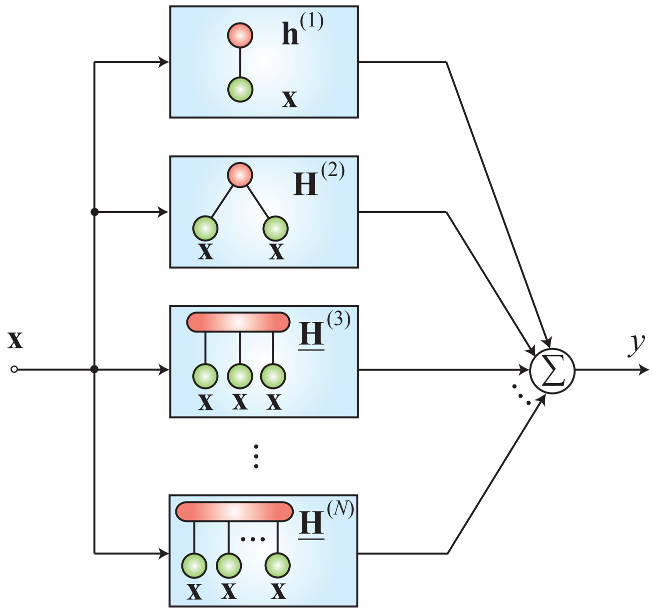


Figure 1.11: A Volterra model of a nonlinear system with memory of length M. Each block computes the tensor product between an nth-order Volterra kernel, $\underline{\mathbf{H}}^{(n)}$, and the vector $\mathbf{x}$ of length M, which comprises M samples of the input signal. The system identification task amounts to estimating the Volterra kernels, $\underline{\mathbf{H}}^{(n)}$, directly or in suitable tensor network formats.

where h_0 is a constant and $H_n(x(t))$ is the nth-order Volterra operator, defined as a generalised convolution of the integral *Volterra kernels* $h^{(n)}(\tau_1, \ldots, \tau_n)$ and the input signal, that is

$$H_n(x(t)) = \int h^{(n)}(\tau_1, \ldots, \tau_n) x(t - \tau_1) \cdots x(t - \tau_n) d\tau_1 \cdots d\tau_n \, . \quad (1.52)$$

The system, which is assumed to be time-invariant and continuous, is treated as a black box, and needs to be represented by appropriate Volterra operators.

In practice, for a finite duration sample input data, $\mathbf{x}$, the discrete system can be modelled using truncated Volterra kernels of size $M \times$

$M \times \cdots \times M$, given by

$$\begin{aligned} H_n(\mathbf{x}) &= \sum_{i_1=1}^{I} \cdots \sum_{i_n=1}^{I} h^{(n)}_{i_1,\ldots,i_n} x_{i_1} \ldots x_{i_2} \\ &= \underline{\mathbf{H}}^{(n)} \bar{\times}_1 \mathbf{x} \, \bar{\times}_2 \mathbf{x} \cdots \bar{\times}_n \mathbf{x}. \end{aligned} \tag{1.53}$$

For simplicity, the Volterra kernels $\underline{\mathbf{H}}^{(n)} = [h^{(n)}_{i_1,\ldots,i_n}]$ are assumed to have the same size M in each mode, and, therefore, to yield a symmetric tensor. Otherwise, they can be symmetrized.

Curse of dimensionality. The output which corresponds to the input $\mathbf{x}$ is written as a sum of N tensor products (see in Figure 1.11), given by

$$y = h_0 + \sum_{n=1}^{N} \underline{\mathbf{H}}^{(n)} \bar{\times}_1 \mathbf{x} \, \bar{\times}_2 \mathbf{x} \cdots \bar{\times}_n \mathbf{x}. \tag{1.54}$$

Despite the symmetry of the Volterra kernels, $\underline{\mathbf{H}}^{(n)}$, the number of actual coefficients of the nth-order kernel to be estimated is still huge, especially for higher-order kernels, and is given by $\frac{(M+n-1)!}{n!(M-1)!}$. As a consequence, the estimation requires a large number of measures (data samples), so that the method for a raw tensor format is only feasible for systems with a relatively small memory and low-dimensional input signals.

1.9.2 Separable Representation of Volterra Kernel

In order to deal with the curse of dimensionality in Volterra kernels, we consider the kernel $\underline{\mathbf{H}}^{(n)}$ to be separable, i.e., it can be expressed in some low rank tensor format, e.g., as a CP tensor or in any other suitable tensor network format (for the concept of general separability of variables, see Part 1).

Volterra-CP model. The first and simplest separable Volterra model, proposed in (Favier *et al.*, 2012), represents the kernels by symmetric tensors of rank R_n in the CP format, that is

$$\underline{\mathbf{H}}^{(n)} = \underline{\mathbf{I}} \times_1 \mathbf{A}_n \times_2 \mathbf{A}_n \cdots \times_n \mathbf{A}_n \,. \tag{1.55}$$

For this tensor representation, the identification problem simplifies into the estimation of N factor matrices, $\mathbf{A}_n$, of size $M \times R_n$ and an offset,

h_0, so that the number of parameters reduces to $M \sum_n R_n + 1$ (note that $R_1 = 1$). Moreover, the implementation of the Volterra model becomes

$$y_k = h_0 + \sum_{n=1}^{N} (\mathbf{x}_k^{\mathrm{T}} \, \mathbf{A}_n)^{\cdot n} \, \mathbf{1}_{R_n} \,, \tag{1.56}$$

where $\mathbf{x}_k = [x_{k-M+1}, \ldots, x_{k-1}, x_k]^{\mathrm{T}}$ comprises M samples of the input signal, and $(\bullet)^{\cdot n}$ represents the element-wise power operator. The entire output vector $\mathbf{y}$ can be computed in a simpler way through the convolution of the input vector $\mathbf{x}$ and the factor matrices $\mathbf{A}_n$, as (Batselier *et al.*, 2016a)

$$\mathbf{y} = h_0 + \sum_{n=1}^{N} (\mathbf{x} * \mathbf{A}_n)^{\cdot n} \, \mathbf{1}_{R_n} \,. \tag{1.57}$$

Volterra-TT model. Alternatively, the Volterra kernels, $\underline{\mathbf{H}}^{(n)}$, can be represented in the TT-format, as

$$\underline{\mathbf{H}}^{(n)} = \langle\!\langle \underline{\mathbf{G}}_n^{(1)}, \underline{\mathbf{G}}_n^{(2)}, \ldots, \underline{\mathbf{G}}_n^{(n)} \rangle\!\rangle \,. \tag{1.58}$$

By exploiting the fast contraction over all modes between a TT-tensor and $\mathbf{x}_k$, we have

$$\underline{\mathbf{H}}^{(n)} \bar{\times} \mathbf{x}_k = (\underline{\mathbf{G}}_n^{(1)} \bar{\times}_2 \mathbf{x}_k)(\underline{\mathbf{G}}_n^{(2)} \bar{\times}_2 \mathbf{x}_k) \cdots (\underline{\mathbf{G}}_n^{(n)} \bar{\times}_2 \mathbf{x}_k) \,.$$

The output signal, can be then computed through the convolution of the core tensors and the input vector, as

$$y_k = h_0 + \sum_{n=1}^{N} \underline{\mathbf{Z}}_{n,1}(1,k,:) \, \underline{\mathbf{Z}}_{n,2}(:,k,:) \, \cdots \, \underline{\mathbf{Z}}_{n,n-1}(:,k,:) \underline{\mathbf{Z}}_{n,n}(:,k) \,,$$

where $\underline{\mathbf{Z}}_{n,m} = \underline{\mathbf{G}}_n^{(m)} *_2 \mathbf{x}$ is a mode-2 partial convolution of the input signal $\mathbf{x}$ and the core tensor $\underline{\mathbf{G}}_n^{(m)}$, for $m = 1, \ldots, n$. A similar method, but with only one TT-tensor, is considered in (Batselier *et al.*, 2016b).

1.9.3 Volterra-based Tensorization for Nonlinear Feature Extraction

Consider nonlinear feature extraction in a supervised learning system, such that the extracted features maximize the Fisher score (Kumar *et al.*, 2009). In other words, for a data sample $\mathbf{x}_k$, which can be a recorded signal in one trial or a vectorization of an image, a feature extracted from $\mathbf{x}_k$ by a nonlinear process is denoted by $y_k = f(\mathbf{x}_k)$. Such constrained (discriminant) feature extraction can be treated as a maximization of the Fisher score

$$\max \frac{\sum_c (\bar{y}_c - \bar{y})^2}{\sum_k (y_k - \bar{y}_{c_k})^2}\,, \tag{1.59}$$

where $\bar{y}_{c_k}$ is the mean feature of the samples in class-k, and $\bar{y}$ the mean feature of all the samples.

Next, we model the nonlinear system $f(\mathbf{x})$ by a truncated Volterra series representation

$$y_k = \sum_{n=1}^{N} \underline{\mathbf{H}}^{(n)} \bar{\times} (\mathbf{x}_k \circ \mathbf{x}_k \circ \cdots \circ \mathbf{x}_k) = \mathbf{h}^{\mathrm{T}}\, \mathbf{z}_k\,, \tag{1.60}$$

where $\mathbf{h}$ and $\mathbf{x}_k$ are vectors comprising all coefficients of the Volterra kernels and

$$\begin{aligned}
\mathbf{h} &= [\mathrm{vec}\left(\mathbf{H}^{(1)}\right)^{\mathrm{T}}, \mathrm{vec}\left(\mathbf{H}^{(2)}\right)^{\mathrm{T}}, \ldots, \mathrm{vec}\left(\mathbf{H}^{(N)}\right)^{\mathrm{T}}]^{\mathrm{T}}\,,\\
\mathbf{z}_k &= [\mathbf{x}_k^{\mathrm{T}}, (\mathbf{x}_k^{\otimes 2})^{\mathrm{T}}, \ldots, (\mathbf{x}_k^{\otimes N})^{\mathrm{T}}]^{\mathrm{T}}\,.
\end{aligned}$$

The shorthand $\mathbf{x}^{\otimes n} = \mathbf{x} \otimes \mathbf{x} \otimes \cdots \otimes \mathbf{x}$ represents the Kronecker product of n vectors $\mathbf{x}$. The offset coefficient, h_0, is omitted in the above Volterra model because it will be eliminated in the objective function (1.59). The vector $\mathbf{h}$ can be shortened by keeping only distinct coefficients, due to symmetry of the Volterra kernels. The augmented sample $\mathbf{z}_k$ needs a similar adjustment but multiplied with the number of occurrences.

Observe that the nonlinear feature extraction, $f(\mathbf{x}_k)$, becomes a linear mapping, as in (1.60) after $\mathbf{x}_k$ is tensorized into $\mathbf{z}_k$. Hence, the nonlinear discriminant in (1.59) can be rewritten in the form of a standard linear discriminant analysis

$$\max \frac{\mathbf{h}^{\mathrm{T}} \mathbf{S}_b \mathbf{h}}{\mathbf{h}^{\mathrm{T}} \mathbf{S}_w \mathbf{h}}\,, \tag{1.61}$$

where $\mathbf{S}_b = \sum_c (\bar{\mathbf{z}}_c - \bar{\mathbf{z}})(\bar{\mathbf{z}}_c - \bar{\mathbf{z}})^{\mathrm{T}}$ and $\mathbf{S}_w = \sum_k (\mathbf{z}_k - \bar{\mathbf{z}}_{c_k})(\mathbf{z}_k - \bar{\mathbf{z}}_{c_k})^{\mathrm{T}}$ are respectively between- and within-scattering matrices of $\mathbf{z}_k$. The problem then boils down to finding generalised principal eigenvectors of $\mathbf{S}_b$ and $\mathbf{S}_w$.

Efficient implementation. The problem with the above analysis is that the length of eigenvectors, $\mathbf{h}$, in (1.61) grows exponentially with the data size, especially for higher-order Volterra kernels. To this end, Kumar *et al.* (2009) suggested to split the data into small patches. Alternatively, we can impose low rank-tensor structures, e.g., the CP or TT format, onto the Volterra kernels, $\mathbf{H}^{(n)}$, or the entire vector $\mathbf{h}$.

1.10 Low-rank Tensor Representations of Sinusoid Signals and their Applications to BSS and Harmonic Retrieval

Harmonic signals are fundamental in many practical applications. This section addresses low-rank structures of sinusoid signals under several tensorization methods. These properties can then be exploited in the blind separation of sinusoid signals or their modulated variants, e.g., the exponentially decaying signals, the examples of which are

$$x(t) = \sin(\omega\, t + \phi)\,, \qquad x(t) = t \sin(\omega\, t + \phi)\;, \tag{1.62}$$

$$x(t) = \exp(-\gamma t) \sin(\omega\, t + \phi)\,, \qquad x(t) = t \exp(-\gamma t)\,, \tag{1.63}$$

for $t = 1, 2, \ldots, L$, $\omega \neq 0$.

1.10.1 Folding - Reshaping of Sinusoid

Harmonic matrix. The harmonic matrix $\mathbf{U}_{\omega,I}$ is a matrix of size $I \times 2$ defined over the two variables, the angular frequency ω and the folding size I, as

$$\mathbf{U}_{\omega,I} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ \cos(k\omega) & \sin(k\omega) \\ \vdots & \vdots \\ \cos((I-1)\omega) & \sin((I-1)\omega) \end{bmatrix}. \tag{1.64}$$

Two-way folding. A matrix of size $I \times J$, folded from a sinusoid signal $x(t)$ of length $L = IJ$, is of rank-2, and can be decomposed as

$$\mathbf{Y} = \mathbf{U}_{\omega,I}\, \mathbf{S}\, \mathbf{U}^{\mathrm{T}}_{\omega I,J}\,, \tag{1.65}$$

where $\mathbf{S}$ is invariant to the folding size I, depends only on the phase ϕ, and takes the form

$$\mathbf{S} = \begin{bmatrix} \sin(\phi) & \cos(\phi) \\ \cos(\phi) & -\sin(\phi) \end{bmatrix}. \tag{1.66}$$

Three-way folding. A third-order tensor of size $I \times J \times K$, where $I, J, K > 2$, reshaped from a sinusoid signal of length L, can take the form of a multilinear rank-(2,2,2) or rank-3 tensor

$$\underline{\mathbf{Y}} = [\![\underline{\mathbf{H}}; \mathbf{U}_{\omega,I}, \mathbf{U}_{\omega I,J}, \mathbf{U}_{\omega IJ,K}]\!], \tag{1.67}$$

where $\underline{\mathbf{H}} = \underline{\mathbf{G}} \times_3 \mathbf{S}$ is a small-scale tensor of size $2 \times 2 \times 2$, and

$$\underline{\mathbf{G}}(:,:,1) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \underline{\mathbf{G}}(:,:,2) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{1.68}$$

The above expression can be derived by folding the signal $y(t)$ two times. We can prove by contradiction that the so-created core tensor $\underline{\mathbf{G}}$ does not have rank-2, but has the following rank-3 tensor representation

$$\underline{\mathbf{G}} = \frac{1}{2}\begin{bmatrix}1\\1\end{bmatrix} \circ \begin{bmatrix}1\\1\end{bmatrix} \circ \begin{bmatrix}1\\1\end{bmatrix} - \frac{1}{2}\begin{bmatrix}-1\\1\end{bmatrix} \circ \begin{bmatrix}-1\\1\end{bmatrix} \circ \begin{bmatrix}-1\\1\end{bmatrix} + 2\begin{bmatrix}0\\1\end{bmatrix} \circ \begin{bmatrix}0\\1\end{bmatrix} \circ \begin{bmatrix}-1\\0\end{bmatrix}.$$

Hence, $\underline{\mathbf{Y}}$ is also a rank-3 tensor. Note that $\underline{\mathbf{Y}}$ does not have a unique rank-3 decomposition.

Remark 1.2. The Tucker-3 decomposition in (1.67) has a fixed core tensor $\underline{\mathbf{G}}$, while the factor matrices are identical for signals of the same frequency.

Higher-order folding - TT-representation. An Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$, where $I_n \geqslant 2$, which is reshaped from a sinusoid signal, can be represented by a multilinear rank-(2,2,..., 2) tensor

$$\underline{\mathbf{Y}} = [\![\underline{\mathbf{H}}; \mathbf{U}_{\omega,I_1}, \mathbf{U}_{\omega J_1,I_2}, \ldots, \mathbf{U}_{\omega J_{N-1},I_N}]\!], \tag{1.69}$$

where $\underline{\mathbf{H}} = \langle\!\langle \underbrace{\underline{\mathbf{G}}, \underline{\mathbf{G}}, \ldots, \underline{\mathbf{G}}}_{(N-2)\text{terms}}, \mathbf{S} \rangle\!\rangle$ is an Nth-order tensor of size $2 \times 2 \times \cdots \times 2$, and $J_n = \prod_{k=1}^{n} I_k$.

Remark 1.3 (TT-representation)**.** Since the tensor $\underline{\mathbf{H}}$ has TT-rank of (2,2,...,2), the folding tensor $\underline{\mathbf{Y}}$ is also a tensor in TT-format of rank-(2,2,...,2), that is

$$\underline{\mathbf{Y}} = \langle\langle \underline{\mathbf{A}}_1, \underline{\mathbf{A}}_2, \ldots, \underline{\mathbf{A}}_N \rangle\rangle, \tag{1.70}$$

where $\underline{\mathbf{A}}_1 = \mathbf{U}_{\omega,I_1}$, $\underline{\mathbf{A}}_N = \mathbf{S}\mathbf{U}^{\mathrm{T}}_{\omega J_{N-1},I_N}$ and $\underline{\mathbf{A}}_n = \underline{\mathbf{G}} \times_2 \mathbf{U}_{\omega J_{n-1},I_n}$ for $n = 2, \ldots, N-1$.

Remark 1.4 (QTT-Tucker representation)**.** When the folding sizes $I_n = 2$, for $n = 1, \ldots, N$, the representation of the folding tensor $\underline{\mathbf{Y}}$ in (1.69) is also known as the *QTT-Tucker format*, given by

$$\underline{\mathbf{Y}} = [\![\underline{\mathbf{H}}; \mathbf{A}_1, \ldots, \mathbf{A}_{N-1}, \mathbf{A}_N]\!], \tag{1.71}$$

where $\mathbf{A}_n = \begin{bmatrix} 1 & 0 \\ \cos(2^{n-1}\omega) & \sin(2^{n-1}\omega) \end{bmatrix}$.

Example 1.9. Separation of damped sinusoid signals.

This example demonstrates the use of multiway folding in a single channel separation of damped sinusoids. We considered a vector composed of P damped sinusoids,

$$y(t) = \sum_{p=1}^{P} a_p \, x_p(t) + n(t)\,, \tag{1.72}$$

where

$$x_p(t) = \exp\left(\frac{-5t}{Lp}\right) \sin\left(\frac{2\pi f_p}{f_s} t + \frac{(p-1)\pi}{P}\right),$$

with frequencies $f_p = 10$, 12 and 14 Hz, and the sampling frequency $f_s = 10 f_P$. Additive Gaussian noise, $n(t)$, was generated at a specific signal-noise-ratio (SNR). The weights, a_p, were set such that the component sources were equally contributing to the mixture, i.e., $a_1\|\mathbf{x}_1\| = \cdots = a_P\|\mathbf{x}_P\|$, and the signal length was $L = 2^d P^2$.

In order to separate the three signals $x_p(t)$ from the mixture $y(t)$, we tensorized the mixture to a dth-order tensor of size $2R \times 2 \times \cdots \times 2 \times 2R$. Under this tensorization, the exponentially decaying signals $\exp(\gamma t)$ yielded rank-1 tensors, while according to (1.69) the sinusoids have

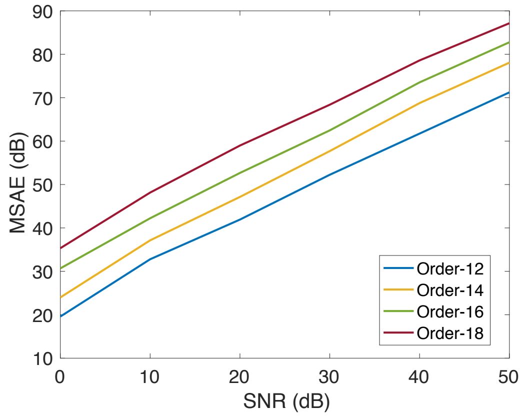


Figure 1.12: Comparison of the mean SAEs for various noise levels SNR, signal lengths, and tensor orders.

TT-representations of rank-$(2, 2, \ldots, 2)$. Hence, the tensors of $x(t)$ can also be represented by tensors in the TT-format of rank-$(2, 2, \ldots, 2)$. We were, therefore, able to approximate $\underline{\mathbf{Y}}$ as a sum of P TT-tensors $\underline{\mathbf{X}}_r$ of rank-$(2, 2, \ldots, 2)$, that is, through the minimization (Phan *et al.*, 2016)

$$\min \quad \|\underline{\mathbf{Y}} - \underline{\mathbf{X}}_1 - \underline{\mathbf{X}}_2 - \cdots - \underline{\mathbf{X}}_P\|_F^2 \,. \tag{1.73}$$

For this purpose, a tensor $\underline{\mathbf{X}}_p$ in a TT-format was fitted sequentially to the residual $\underline{\mathbf{Y}}_p = \underline{\mathbf{Y}} - \sum_{s \neq p} \underline{\mathbf{X}}_s$, calculated by the difference between the data tensor $\underline{\mathbf{Y}}$ and its approximation by the other TT-tensors $\underline{\mathbf{X}}_s$ where $s \neq p$, that is,

$$\arg\min_{\underline{\mathbf{X}}_p} \|\underline{\mathbf{Y}}_p - \underline{\mathbf{X}}_p\|_F^2 \,, \tag{1.74}$$

for $p = 1, \ldots, P$. Figure 1.12 illustrates the mean SAEs (MSAE) of the estimated signals for various noise levels SNR $= 0,\ 10,\ \ldots,\ 50$ dB, and different signal lengths $K = 9 \times 2^d$, where $d = 12, 14, 16, 18$.

On average, *an improvement of 2 dB SAE is achieved if the signal is two times longer.* If the signal has less than $L = 9 \times 2^6 = 576$ samples, the estimation quality will deteriorate by about 12 dB compared to the case when signal length of $L = 9 \times 2^{12}$. For such cases, we suggest to augment the signals using other tensorizations before performing the source extraction, e.g., by construction of multiway Toeplitz or Hankel tensors. Example 1.10 further illustrates the separation of short length signals.

1.10.2 Toeplitz Matrix and Toeplitz Tensors of Sinusoidal Signals

Toeplitz matrix of sinusoid. The Toeplitz matrix, $\mathbf{Y}$, of a sinusoid signal, $y(t) = \sin(\omega\, t + \phi)$, is of rank-2 and can be decomposed as

$$\mathbf{Y} = \begin{bmatrix} y(1) & y(2) \\ y(2) & y(3) \\ \vdots & \vdots \\ y(I) & y(I+1) \end{bmatrix} \mathbf{Q}_T \begin{bmatrix} y(I) & \cdots & y(L) \\ y(I-1) & \cdots & y(L-1) \end{bmatrix}, \tag{1.75}$$

where $\mathbf{Q}_T$ is invariant to the selection of folding length I, and has the form

$$\mathbf{Q}_T = \frac{1}{\sin^2(\omega)} \begin{bmatrix} -y(3) & y(2) \\ y(2) & -y(1) \end{bmatrix}. \tag{1.76}$$

The above expression follows from the fact that

$$\begin{bmatrix} y(i) & y(i+1) \end{bmatrix} \begin{bmatrix} -y(3) & y(2) \\ y(2) & -y(1) \end{bmatrix} \begin{bmatrix} y(j) \\ y(j-1) \end{bmatrix} = \sin^2(\omega)\, y(j-i+1)\,.$$

Toeplitz tensor of sinusoid. An Nth-order Toeplitz tensor, tensorized from a sinusoidal signal, has a TT-Tucker representation

$$\underline{\mathbf{Y}} = [\![\underline{\mathbf{G}}; \mathbf{U}_1, \ldots, \mathbf{U}_{N-1}, \mathbf{U}_N]\!] \tag{1.77}$$

where the factor matrices $\mathbf{U}_n$ are given by

$$\mathbf{U}_1 = \begin{bmatrix} y(1) & y(2) \\ \vdots & \vdots \\ y(J_1) & y(J_1+1) \end{bmatrix}, \quad \mathbf{U}_N = \begin{bmatrix} y(J_{N-1}-1) & y(J_{N-1}-2) \\ \vdots & \vdots \\ y(L) & y(L-1) \end{bmatrix},$$

$$\mathbf{U}_n = \begin{bmatrix} y(J_{n-1}) & y(J_{n-1}+1) \\ \vdots & \vdots \\ y(J_n-1) & y(J_n) \end{bmatrix}, \quad n = 2, \ldots, N-1\,, \tag{1.78}$$

in which $J_n = I_1 + I_2 + \cdots + I_n$. The core tensor $\underline{\mathbf{G}}$ is an Nth-order tensor of size $2 \times 2 \times \cdots \times 2$, in a TT-format, given by

$$\underline{\mathbf{G}} = \langle\!\langle \underline{\mathbf{G}}^{(1)}, \underline{\mathbf{G}}^{(2)}, \ldots, \underline{\mathbf{G}}^{(N-1)} \rangle\!\rangle, \tag{1.79}$$

where $\underline{\mathbf{G}}^{(1)} = \mathbf{T}(1)$ is a matrix of size $1 \times 2 \times 2$, while the core tensors $\underline{\mathbf{G}}^{(n)}$, for $n = 2, \ldots, N-1$, are of size $2 \times 2 \times 2$ and have two horizontal slices, given by

$$\underline{\mathbf{G}}^{(n)}(1,:,:) = \mathbf{T}(J_{n-1} - n + 2)\,, \quad \underline{\mathbf{G}}^{(n)}(2,:,:) = \mathbf{T}(J_{n-1} - n + 1)\,,$$

with

$$\mathbf{T}(I) = \frac{1}{\sin^2(\omega)} \begin{bmatrix} -y(I+2) & y(I+1) \\ y(I+1) & -y(I) \end{bmatrix}. \tag{1.80}$$

Following the two-stage Toeplitz tensorization, and upon applying (1.75), we can deduce the decomposition in (1.77) from that for the $(N-1)$th-order Toeplitz tensor.

Remark 1.5. For second-order tensorization, the core tensor $\underline{\mathbf{G}}$ in (1.79) comprises only $\mathbf{G}^{(1)}$, which is identical to the matrix $\mathbf{Q}_T$ in (1.76).

Quantized Toeplitz tensor. An $(L-1)$th-order Toeplitz tensor of a sinusoidal signal of length L and size $2 \times 2 \times \cdots \times 2$ has a TT-representation with $(L-3)$ identical core tensors $\underline{\mathbf{G}}$, in the form

$$\underline{\mathbf{Y}} = \langle\!\langle \underline{\mathbf{G}}, \underline{\mathbf{G}}, \ldots, \underline{\mathbf{G}}, \begin{bmatrix} y(L-1) & y(L) \\ y(L-2) & y(L-1) \end{bmatrix} \rangle\!\rangle,$$

where

$$\underline{\mathbf{G}}(1,:,:) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \underline{\mathbf{G}}(2,:,:) = \begin{bmatrix} 0 & 1 \\ -1 & 2\cos(\omega) \end{bmatrix}.$$

Example 1.10. Separation of short-length damped sinusoid signals.

This example illustrates the use of Toeplitz-based tensorization in the separation of damped sinusoid signals from a short-length observation. We considered a single signal composed by $P = 3$ damped

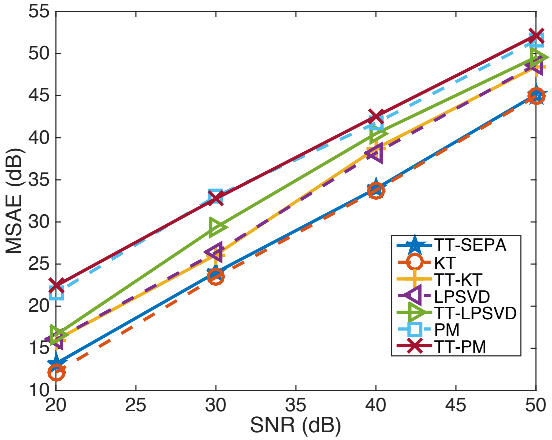


Figure 1.13: Mean SAEs (MSAE) of the estimated signals in Example 1.10, for various noise levels SNR.

sinusoids of length $L = 66$, given by

$$y(t) = \sum_{p=1}^{P} a_p \, x_p(t) + n(t) \,, \tag{1.81}$$

where

$$x(t) = \exp(\frac{-pt}{30}) \, \sin(\frac{2\pi f_p}{f_s} t + \frac{p\pi}{7}) \tag{1.82}$$

with frequencies $f_p = 10$, 11 and 12 Hz, the sampling frequency $f_s =$ 300 Hz, and the mixing factors $a_p = p$. Additive Gaussian noise $n(t)$ was generated at a specific signal-noise-ratio.

In order to separate the three signals, $x_p(t)$, from the mixture $y(t)$, we first tensorized the observed signal to a 7th-order Toeplitz tensor of size $16 \times 8 \times 8 \times 8 \times 8 \times 8 \times 16$, then folded this tensor to a 23th-order tensor of size $2 \times 2 \times \cdots \times 2$. With this tensorization, according to (1.77) and (1.69), each damped sinusoid $x_p(t)$ had a TT-representation of rank-$(2, 2, \ldots, 2)$. The result produced by minimizing the cost function (1.73), annotated by TT-SEPA, is shown in Figure 1.13 as a solid line with star marker. The so obtained performance was much better than in Example 1.9, even for the signal length of only 66 samples.

We note that the parameters of the damped signals can be estimated using linear self-prediction (auto-regression) methods, e.g., singular value decomposition of the Hankel-type matrix as in the Kumaresan-Tufts (KT) method (Kumaresan and Tufts, 1982). As shown in Figure 1.13, the obtained results based on the TT-decomposition were slightly better than those using the KT method. For this particular problem, the estimation performance can even be higher when applying self-prediction algorithms, which exploit the low-rank structure of damped signals, e.g., TT-KT, and TT-linear prediction methods based on SVD. For a detailed derivation of these algorithms, see (Phan *et al.*, 2017).

1.10.3 Hankel Matrix and Hankel Tensor of Sinusoidal Signal

Hankel tensor of sinusoid. The Hankel tensor of a sinusoid signal $y(t)$ is a TT-Tucker tensor,

$$\underline{\mathbf{Y}} = [\![\underline{\mathbf{G}}; \mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N]\!] \,, \tag{1.83}$$

for which the factor matrices are defined in (1.78). The core tensor $\underline{\mathbf{G}}$ is an Nth-order tensor of size $2 \times 2 \times \cdots \times 2$, in the TT-format, given by

$$\underline{\mathbf{G}} = \langle\!\langle \underline{\mathbf{G}}^{(1)}, \underline{\mathbf{G}}^{(2)}, \ldots, \underline{\mathbf{G}}^{(N-1)} \rangle\!\rangle \,, \tag{1.84}$$

where $\underline{\mathbf{G}}^{(1)} = \mathbf{H}(J_1)$ is a matrix of size $1 \times 2 \times 2$, while the core tensors $\underline{\mathbf{G}}^{(n)}$, for $n = 2, \ldots, N-1$, are of size $2 \times 2 \times 2$ and have two horizontal slices, given by

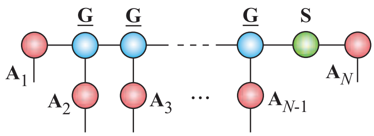
$$\underline{\mathbf{G}}^{(n)}(1, :, :) = \mathbf{H}(J_n - n + 1) \,, \quad \underline{\mathbf{G}}^{(n)}(2, :, :) = \mathbf{H}(J_n - n + 2) \,,$$

with

$$\mathbf{H}(I) = \frac{1}{\sin^2(\omega)} \begin{bmatrix} y(I) & -y(I+1) \\ -y(I-1) & y(I) \end{bmatrix} . \tag{1.85}$$

Remark 1.6. The two TT-Tucker representations of the Toeplitz and Hankel tensors of the same sinusoid have similar factor matrices $\mathbf{U}_n$, but their core tensors are different.

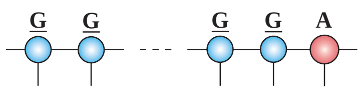
1. Folded tensor



$$\underline{\mathbf{G}} = \left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\right]$$

$$\mathbf{S} = \begin{bmatrix}\sin(\phi) & \cos(\phi)\\ \cos(\phi) & -\sin(\phi)\end{bmatrix}, \ \mathbf{A}_n = \mathbf{U}_{\omega,2^{n-1}}$$

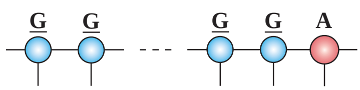
2. Toeplitz tensor



$$\underline{\mathbf{G}} = \left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1\\ -1 & 2\cos(\omega)\end{bmatrix}\right]$$

$$\mathbf{A} = \begin{bmatrix}y(L-1) & y(L)\\ y(L-2) & y(L-1)\end{bmatrix}$$

3. Hankel tensor



$$\underline{\mathbf{G}} = \left[\begin{bmatrix}2\cos(\omega) & -1\\ 1 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right]$$

$$\mathbf{A} = \begin{bmatrix}y(L-2) & y(L-1)\\ y(L-1) & y(L)\end{bmatrix}$$

Figure 1.14: Representations of a sinusoid signal in different quantized tensor formats of size $2 \times 2 \times \cdots \times 2$.

Quantized Hankel tensor. An $(L-1)$th-order Hankel tensor of size $2 \times 2 \times \cdots \times 2$ of a sinusoid signal of length L has a TT-representation with $(N-2)$ identical core tensors $\underline{\mathbf{G}}$, in the form

$$\underline{\mathbf{Y}} = \langle\!\langle \underline{\mathbf{G}}, \underline{\mathbf{G}}, \ldots, \underline{\mathbf{G}}, \begin{bmatrix}y(L-2) & y(L-1)\\ y(L-1) & y(L)\end{bmatrix} \rangle\!\rangle,$$

where

$$\underline{\mathbf{G}}(1,:,:) = \begin{bmatrix}2\cos(\omega) & -1\\ 1 & 0\end{bmatrix}, \quad \underline{\mathbf{G}}(2,:,:) = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

Finally, representations of the sinusoid signal in various tensor format of size are summarised in Figure 1.14.

1.11 Summary

This chapter has introduced several common tensorization methods, together with their properties and illustrative applications in blind source separation, blind identification, denoising, and harmonic retrieval. The main criterion for choosing a suitable tensorization is that the tensor generated from lower-order original data must reveal the underlying low-rank tensor structure in some tensor format. For example, the folded tensors of mixtures of damped sinusoid signals have low-rank QTT representation, while the derivative tensors in blind identification admit the CP decomposition. The Toeplitz and Hankel tensor foldings augment the number of signal entries, through the replication of signal segments (redundancy), and in this way become suited to modeling of signals of short length. A property crucial to the solution via the tensor networks shown in this chapter, is that the tensors can be generated in the TT/QTT format, if the generating vector admits a low-rank QTT representation.

In modern data analytics problems, such as regression and deep learning, the number of model parameters can be huge, which renders the model intractable. Tensorization can then serve as a remedy, by representing the parameters in some low-rank tensor format. For further discussion on tensor representation of parameters in tensor regression, we refer to Chapter 2. A wide class of optimization problems including of solving linear systems, eigenvalue decomposition, singular value decomposition, Canonical Correlation Analysis (CCA) are addressed in Chapter 3. The tensor structures for Boltzmann machines and convolutional deep neural networks (CNN) are provided in Chapter 4.

References

P.-A. Absil and I. V. Oseledets. Low-rank retractions: A survey and new results. *Computational Optimization and Applications*, 62(1):5–29, 2015. URL http://dx.doi.org/10.1007/s10589-014-9714-4.

P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

P.-A. Absil, R. Mahony, and J. Trumpf. An extrinsic look at the Riemannian Hessian. In *Geometric Science of Information*, pages 361–368. Springer, 2013.

R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA Journal of Numerical Analysis*, 22(3):359–390, 2002.

G. I. Allen and M. Maletic-Savatic. Sparse non-negative generalized PCA with applications to metabolomics. *Bioinformatics*, 27 (21):3029–3035, 2011.

C. A. Andersson and R. Bro. The N-way toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems*, 52(1):1–4, 2000. URL http://www.models.life.ku.dk/source/nwaytoolbox/.

F. Anselmi, L. Rosasco, C. Tan, and T. Poggio. Deep convolutional networks are hierarchical kernel machines. *arXiv preprint arXiv:1508.01084*, 2015.

M. August, M. Bañuls, and T. Huckle. On the approximation of functionals of very large hermitian matrices represented as matrix product operators. *CoRR*, abs/1610.06086, 2016. URL http://arxiv.org/abs/1610.06086.

F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical report, Department of Statistics, University of California, Berkeley, 2005.

M. Bachmayr and W. Dahmen. Adaptive near-optimal rank tensor approximation for high-dimensional operator equations. *Foundations of Computational Mathematics*, 15(4):839–898, 2015.

M. Bachmayr and R. Schneider. Iterative methods based on soft thresholding of hierarchical tensors. *Foundations of Computational Mathematics*, pages 1–47, 2016.

M. Bachmayr, R. Schneider, and A. Uschmajew. Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations. *Foundations of Computational Mathematics*, 16(6):1423–1472, 2016.

B. W. Bader and T. G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32(4):635–653, 2006.

B. W. Bader and T. G. Kolda. MATLAB tensor toolbox version 2.6, February 2015. URL http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/.

J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, 2013.

K. Batselier, Z. Chen, H. Liu, and N. Wong. A tensor-based volterra series black-box nonlinear system identification and simulation framework. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7, 2016a.

K. Batselier, Z. Chen, and N. Wong. Tensor train alternating linear scheme for MIMO Volterra system identification. *CoRR*, abs/1607.00127, 2016b. URL http://arxiv.org/abs/1607.00127.

P. Benner, V. Khoromskaia, and B. N. Khoromskij. A reduced basis approach for calculation of the Bethe–Salpeter excitation energies by using low-rank tensor factorisations. *Molecular Physics*, 114(7-8):1148–1161, 2016.

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

G. C. Bento, O. P. Ferreira, and J. G. Melo. Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *arXiv preprint arXiv:1609.04869*, 2016.

G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *arXiv preprint arXiv:1611.09347*, 2016.

M. Billaud-Friess, A. Nouy, and O. Zahm. A tensor approximation method based on ideal minimal residual formulations for the solution of high-dimensional problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(6):1777–1806, 2014.

S. A. Billings. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains.* John Wiley & Sons, 2013.

D. Bini. Toeplitz matrices, algorithms and applications. *ECRIM News Online Edition,*, 3(22):852–872, 1995.

S. K. Biswas and P. Milanfar. Linear support tensor machine: Pedestrian detection in thermal infrared images. *arXiv preprint arXiv:1609.07878*, 2016.

T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

M. Bolten, K. Kahl, and S. Sokolović. Multigrid Methods for Tensor Structured Markov Chains with Low Rank Approximation. *SIAM Journal on Scientific Computing*, 38(2):A649–A667, 2016. URL http://adsabs.harvard.edu/abs/2016arXiv160506246B.

S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 406–414. MIT, 2011.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a MATLAB toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(1):1455–1459, 2014.

M. Boussé, O. Debals, and L. De Lathauwer. A tensor-based method for large-scale blind source separation using segmentation. Technical Report Tech. Report 15-59, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2015, submitted.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

R. Bro. Multiway calibration. Multilinear PLS. *Journal of Chemometrics*, 10 (1):47–61, 1996.

R. J. Bursill, T. Xiang, and G. A. Gehring. The density matrix renormalization group for a quantum spin chain at non-zero temperature. *Journal of Physics: Condensed Matter*, 8(40):L583, 1996.

C. Caiafa and A. Cichocki. Computing sparse representations of multidimensional signals using Kronecker bases. *Neural Computaion*, 25(1):186–220, 2013.

C. Caiafa and A. Cichocki. Stable, robust, and super–fast reconstruction of tensors using multi-way projections. *IEEE Transactions on Signal Processing*, 63(3):780–793, 2015.

P. Calabrese and J. Cardy. Entanglement entropy and quantum field theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(06):P06002, 2004.

L. Cambier and P.-A. Absil. Robust low-rank matrix completion by Riemannian optimization. *SIAM Journal on Scientific Computing*, 38(5):S440–S460, 2016.

E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6): 877–905, 2008.

A. B. Chan, N. Vasconcelos, and P. J. Moreno. A family of probabilistic kernels based on information divergence. Technical report, University of California, San Diego, CA, SVCL-TR-2004-1, 2004.

R. Chatterjee and T. Yu. Generalized coherent states, reproducing kernels, and quantum support vector machines. *arXiv preprint arXiv:1612.03713*, 2016.

C. Chen, J. Huang, L. He, and H. Li. Preconditioning for accelerated iteratively reweighted least squares in structured sparsity reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

H. Chen. *Structured Tensors: Theory and Applications*. PhD thesis, The Hong Kong Polytechnic University, Hong Kong, 2016.

J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang. On the equivalence of Restricted Boltzmann Machines and Tensor Network States. *ArXiv e-prints*, 2017.

S. Chen and S. A. Billings. Representations of non-linear systems: the narmax model. *International Journal of Control*, 49(3):1013–1032, 1989. URL http://www.tandfonline.com/doi/abs/10.1080/00207178908559683.

Z. Chen, K. Batselier, J. A. K. Suykens, and N. Wong. Parallelized tensor train learning of polynomial classifiers. *CoRR*, abs/1612.06505, 2016. URL http://arxiv.org/abs/1612.06505.

H. Cho, D. Venturi, and G. E. Karniadakis. Numerical methods for high-dimensional probability density function equations. *Journal of Computational Physics*, 305:817–837, 2016.

J. H. Choi and S. Vishwanathan. DFacTo: Distributed factorization of tensors. In *Advances in Neural Information Processing Systems*, pages 1296–1304, 2014.

D. Chu, L.-Z. Liao, M. K. Ng, and X. Zhang. Sparse canonical correlation analysis: New formulation and algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):3050–3065, 2013.

A. Cichocki. Tensor decompositions: New concepts in brain data analysis? *Journal of the Society of Instrument and Control Engineers*, 50(7):507–516, 2011.

A. Cichocki. Era of big data processing: A new approach via tensor networks and tensor decompositions, (invited). In *Proceedings of the International Workshop on Smart Info-Media Systems in Asia (SISA2013)*, September 2013. URL http://arxiv.org/abs/1403.2048.

A. Cichocki. Tensor networks for big data analytics and large-scale optimization problems. *arXiv preprint arXiv:1407.3124*, 2014.

A. Cichocki and R. Zdunek. Multilayer nonnegative matrix factorisation. *Electronics Letters*, 42(16):1, 2006.

A. Cichocki and R. Zdunek. Regularized alternating least squares algorithms for non-negative matrix/tensor factorization. In *International Symposium on Neural Networks*, pages 793–802. Springer, 2007.

A. Cichocki, W. Kasprzak, and S. Amari. Multi-layer neural networks with a local adaptive learning rule for blind separation of source signals. In *Proc. Int. Symposium Nonlinear Theory and Applications (NOLTA), Las Vegas, NV*, pages 61–65. Citeseer, 1995.

A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Chichester, 2009.

A. Cichocki, S. Cruces, and S. Amari. Generalized alpha-beta divergences and their application to rubust nonnegative matrix factorization. *Entropy*, 13: 134–170, 2011.

A. Cichocki, S. Cruces, and S. Amari. Log-determinant divergences revisited: Alpha-beta and gamma log-det divergences. *Entropy*, 17(5):2988–3034, 2015.

A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic. Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

N. Cohen and A. Shashua. Inductive bias of deep convolutional networks through pooling geometry. *CoRR*, abs/1605.06743, 2016. URL http://arxiv.org/abs/1605.06743.

N. Cohen and A. Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 955–963, 2016.

N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *29th Annual Conference on Learning Theory*, pages 698–728, 2016.

P. Comon and M. Rajih. Blind identification of under-determined mixtures based on the characteristic function. *Signal Processing*, 86(9):2271–2281, 2006.

E. Corona, A. Rahimian, and D. Zorin. A Tensor-Train accelerated solver for integral equations in complex geometries. *arXiv preprint arXiv:1511.06029*, November 2015.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, 1995.

J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900, 2015.

C. Da Silva and F. J. Herrmann. Hierarchical Tucker tensor optimization – Applications to tensor completion. In *Proc. 10th International Conference on Sampling Theory and Applications*, volume 1, 2013.

A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

O. Debals and L. De Lathauwer. Stochastic and deterministic tensorization for blind signal separation. In E. Vincent, A. Yeredor, Z. Koldovsky, and P. Tichavský, editors, *Proceedings of the 12th International Conference Latent Variable Analysis and Signal Separation*, pages 3–13. Springer International Publishing, 2015.

O. Debals, M. Van Barel, and L. De Lathauwer. Löwner-based blind signal separation of rational functions with applications. *IEEE Transactions on Signal Processing*, 64(8):1909–1918, 2016a.

O. Debals, M. Sohail, and L. De Lathauwer. Analytical multi-modulus algorithms based on coupled canonical polyadic decompositions. Technical report, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2016b.

S. V. Dolgov. TT-GMRES: Solution to a linear system in the structured tensor format. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 28(2):149–172, 2013.

S. V. Dolgov. *Tensor Product Methods in Numerical Simulation of High-dimensional Dynamical Problems.* PhD thesis, Faculty of Mathematics and Informatics, University Leipzig, Germany, Leipzig, Germany, 2014.

S. V. Dolgov and B. N. Khoromskij. Two-level QTT-Tucker format for optimized tensor calculus. *SIAM Journal on Matrix Analysis and Applications*, 34(2):593–623, 2013.

S. V. Dolgov and B. N. Khoromskij. Simultaneous state-time approximation of the chemical master equation using tensor product formats. *Numerical Linear Algebra with Applications*, 22(2):197–219, 2015.

S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems. *arXiv preprint arXiv:1301.6068*, 2013a.

S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems. *arXiv preprint arXiv:1304.1222*, 2013b.

S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM Journal on Scientific Computing*, 36(5):A2248–A2271, 2014.

S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, and D. V. Savostyanov. Computation of extreme eigenvalues in higher dimensions using block tensor train format. *Computer Physics Communications*, 185(4):1207–1216, 2014.

S. V. Dolgov, J. W. Pearson, D. V. Savostyanov, and M. Stoll. Fast tensor product solvers for optimization problems with fractional differential equations as constraints. *Applied Mathematics and Computation*, 273:604–623, 2016.

J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277, 2010.

A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. Asymptotic behavior of ℓ_p-based Laplacian regularization in semi-supervised learning. *arXiv e-prints arXiv:1603.00564*, March 2016.

D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.

M. Espig, M. Schuster, A. Killaitis, N. Waldren, P. Wähnert, S. Handschuh, and H. Auer. TensorCalculus library, 2012. URL http://gitorious.org/tensorcalculus.

G. Evenbly and G. Vidal. Algorithms for entanglement renormalization. *Physical Review B*, 79(14):144108, 2009.

G. Evenbly and G. Vidal. Tensor network renormalization yields the multiscale entanglement renormalization Ansatz. *Physical Review Letters*, 115 (20):200401, 2015.

G. Evenbly and S. R. White. Entanglement renormalization and wavelets. *Physical Review Letters*, 116(14):140403, 2016a.

G. Evenbly and S. R. White. Representation and design of wavelets using unitary circuits. *arXiv e-prints*, 2016b.

G. Favier, A. Y. Kibangou, and T. Bouilloc. Nonlinear system modeling and identification using Volterra-PARAFAC models. *International Journal of Adaptive Control and Signal Processing*, 26(1), 2012. URL http://dx.doi.org/10.1002/acs.1272.

A. Fischer and C. Igel. An introduction to restricted Boltzmann machines. In *Iberoamerican Congress on Pattern Recognition*, pages 14–36. Springer, 2012.

S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, New York, 2013.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

C. Gao and X.-J. Wu. Kernel support tensor regression. *Procedia Engineering*, 29:3986–3990, 2012.

T. Garipov, D. Podoprikhin, A. Novikov, and D. P. Vetrov. Ultimate tensorization: compressing convolutional and FC layers alike. *CoRR*, abs/1611.03214, 2016.

S. Garreis and M. Ulbrich. Constrained optimization with low-rank tensors and applications to parametric problems with PDEs. *SIAM Journal on Scientific Computing*, 39(1):A25–A54, 2017.

D. F. Gleich, L.-H. Lim, and Y. Yu. Multilinear PageRank. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1507–1541, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016, Cambridge, MA. http://www.deeplearningbook.org.

L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

L. Grasedyck, D. Kessner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36:53–78, 2013.

L. Grasedyck, M. Kluge, and S. Krämer. Variants of alternating least squares tensor completion in the tensor train format. *SIAM Journal on Scientific Computing*, 37(5):A2424–A2450, 2015.

R. M. Gray. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006.

D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.*, 105:150401, 2010. URL http://link.aps.org/doi/10.1103/PhysRevLett.105.150401.

Z. Hao, L. He, B. Chen, and X. Yang. A linear support higher-order tensor machine for classification. *IEEE Transactions on Image Processing*, 22(7): 2911–2920, 2013.

P. D. Hoff. Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics*, 9(3):1169–1193, 2015.

S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2), 2012a.

S. Holtz, T. Rohwedder, and R. Schneider. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731, 2012b.

R. Hosseini and S. Sra. Matrix manifold optimization for Gaussian mixtures. In *Advances in Neural Information Processing Systems*, pages 910–918, 2015.

M. Hou. *Tensor-based Regression Models and Applications*. PhD thesis, University Laval, Quebec, Canada, 2017.

M. Hou, Y. Wang, and B. Chaib-draa. Online local Gaussian processes for tensor-variate regression: Application to fast reconstruction of limb movements from brain signal. In *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5490–5494. IEEE, 2015.

M. Hou, Q. Zhao, B. Chaib-draa, and A. Cichocki. Common and discriminative subspace kernel-based multiblock tensor partial least squares regression. In *Proc. of Thirtieth AAAI Conference on Artificial Intelligence*, 2016a.

J. Z. Huang, H. Shen, and A. Buja. The analysis of two-way functional data using two-way regularized singular value decompositions. *Journal of the American Statistical Association*, 104(488):1609–1620, 2009. URL http://EconPapers.repec.org/RePEc:bes:jnlasa:v:104:i:488:y:2009:p:1609-1620.

R.-Z. Huang, H.-J. Liao, Z.-Y. Liu, H.-D. Xie, Z.-Y. Xie, H.-H. Zhao, J. Chen, and T. Xiang. A generalized Lanczos method for systematic optimization of tensor network states. *ArXiv e-prints*, 2016.

C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf. Strictly single-site DMRG algorithm with subspace expansion. *Physical Review B*, 91(15):155115, 2015.

C. Hubig, I. P. McCulloch, and U. Schollwöck. Generic construction of efficient matrix product operators. *Phys. Rev. B*, 95:035129, 2017. URL http://link.aps.org/doi/10.1103/PhysRevB.95.035129.

T. Huckle and K. Waldherr. Subspace iteration methods in terms of matrix product states. *PAMM*, 12(1):641–642, 2012.

M. Imaizumi and K. Hayashi. Doubly decomposing nonparametric tensor regression. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 727–736, 2016.

K. Inoue, K. Hara, and K. Urahama. Robust multilinear principal component analysis. In *IEEE 12th International Conference on Computer Vision*, pages 591–597. IEEE, 2009.

M. Ishteva, L. De Lathauwer, P. A. Absil, and S. Van Huffel. Differential-geometric Newton method for the best rank-(r_1, r_2, r_3) approximation of tensors. *Numerical Algorithms*, 51(2):179–194, 2009.

M. Ishteva, P. A. Absil, S. Van Huffel, and L. De Lathauwer. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. *SIAM Journal on Matrix Analysis and Applications*, 32(1):115–135, 2011.

I. Jeon, E. Papalexakis, U. Kang, and C. Faloutsos. Haten2: Billion-scale tensor decompositions. In *Proceedings of the 31st IEEE International Conference on Data Engineering (ICDE)*, pages 1047–1058. IEEE, 2015.

I. Jeon, E. E. Papalexakis, C. Faloutsos, L. Sael, and U. Kang. Mining billion-scale tensors: Algorithms and discoveries. *The VLDB Journal*, pages 1–26, 2016.

I. Jolliffe. A note on the use of principal components in regression. *Applied Statistics*, pages 300–303, 1982.

M. H. Kamal, B. Heshmat, R. Raskar, P. Vandergheynst, and G. Wetzstein. Tensor low-rank and sparse light field photography. *Computer Vision and Image Understanding*, 145:172–181, 2016.

U. Kang, E. E. Papalexakis, A. Harpale, and C. Faloutsos. GigaTensor: Scaling tensor analysis up by 100 times – algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pages 316–324, August 2012.

Y.-J. Kao, Y.-D. Hsieh, and P. Chen. Uni10: An open-source library for tensor network algorithms. In *Journal of Physics: Conference Series*, volume 640, page 012040. IOP Publishing, 2015.

L. Karlsson, D. Kressner, and A. Uschmajew. Parallel algorithms for tensor completion in the CP format. *Parallel Computing*, 57:222–234, 2016.

H. Kasai and B. Mishra. Riemannian preconditioning for tensor completion. *arXiv preprint arXiv:1506.02159*, 2015.

V. A. Kazeev, B. N. Khoromskij, and E. E. Tyrtyshnikov. Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity. *SIAM Journal on Scientific Computing*, 35(3), 2013.

V. Khoromskaia and B. N. Khoromskij. Fast tensor method for summation of long-range potentials on 3D lattices with defects. *Numerical Linear Algebra with Applications*, 23(2):249–271, 2016. URL http://dx.doi.org/10.1002/nla.2023.

B. N. Khoromskij. $O(d \log N)$-quantics approximation of N-d tensors in high-dimensional numerical modeling. *Constructive Approximation*, 34(2):257–280, 2011a.

B. N. Khoromskij. Tensors-structured numerical methods in scientific computing: Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 110(1):1–19, 2011b. URL http://www.mis.mpg.de/de/publications/preprints/2010/prepr2010-21.html.

B. N. Khoromskij and S. Miao. Superfast wavelet transform using quantics-TT approximation. I. application to Haar wavelets. *Computational Methods in Applied Mathematics*, 14(4):537–553, 2014. URL http://dx.doi.org/10.1515/cmam-2014-0016.

B. N. Khoromskij and I. Oseledets. Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs. *Computational Methods in Applied Mathematics*, 10(4):376–394, 2010.

B. N. Khoromskij and C. Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM Journal on Scientific Computing*, 33(1):364–385, 2011.

B. N. Khoromskij and A. Veit. Efficient computation of highly oscillatory integrals by using QTT tensor approximation. *Computational Methods in Applied Mathematics*, 16(1):145–159, 2016.

V. Khrulkov and I. Oseledets. Desingularization of bounded-rank matrix sets. *arXiv preprint arXiv:1612.03973*, 2016.

H.-J. Kim, E. Ollila, and V. Koivunen. New robust Lasso method based on ranks. In *23rd European Signal Processing Conference (EUSIPCO)*, pages 699–703. IEEE, 2015.

T. K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, 2009.

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'07)*, pages 1–8, 2007.

O. Koch and C. Lubich. Dynamical low rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29:434–454, 2007.

O. Koch and C. Lubich. Dynamical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2360–2375, 2010.

N. Koep and S. Weichwald. Pymanopt: A Python Toolbox for manifold optimization using automatic differentiation. *arXiv preprint arXiv:1603.03236*, 2016.

E. Kokiopoulou, J. Chen, and Y. Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602, 2011.

D. Kolesnikov and I. V. Oseledets. Convergence analysis of projected fixed-point iteration on a low-rank matrix manifold. *arXiv preprint arXiv:1604.02111*, 2016.

I. Kotsia, W. Guo, and I. Patras. Higher rank support tensor machines for visual recognition. *Pattern Recognition*, 45(12):4192–4203, 2012.

D. Kressner and F. Macedo. Low-rank tensor methods for communicating Markov processes. In *Quantitative Evaluation of Systems*, pages 25–40. Springer, 2014.

D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, 2011a.

D. Kressner and C. Tobler. Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems. *Computational Methods in Applied Mathematics*, 11(3):363–381, 2011b.

D. Kressner and C. Tobler. Algorithm 941: HTucker–A MATLAB toolbox for tensors in hierarchical Tucker format. *ACM Transactions on Mathematical Software*, 40(3):22, 2014.

D. Kressner and A. Uschmajew. On low-rank approximability of solutions to high-dimensional operator equations and eigenvalue problems. *Linear Algebra and its Applications*, 493:556–572, 2016.

D. Kressner, M. Steinlechner, and A. Uschmajew. Low-rank tensor methods with subspace correction for symmetric eigenvalue problems. *SIAM Journal on Scientific Computing*, 36(5):A2346–A2368, 2014a.

D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2): 447–468, 2014b.

D. Kressner, M. Steinlechner, and B. Vandereycken. Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure. *SIAM Journal on Scientific Computing*, 38(4):A2018–A2044, 2016.

R. Kumar, A. Banerjee, and B. C. Vemuri. Volterrafaces: Discriminant analysis using volterra kernels. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 150–155, 2009.

R. Kumaresan and D. W. Tufts. Estimating the parameters of exponentially damped sinusoids and pole-zero modelling in noise. *IEEE Trans. Acoust. Speech Signal Processing*, 30(7):837–840, 1982.

L. De Lathauwer. Algebraic techniques for the blind deconvolution of constant modulus signals. In *The 2004 12th European Signal Processing Conference*, pages 225–228, 2004.

V. Lebedev and V. Lempitsky. Fast convolutional neural networks using group-wise brain damage. *arXiv preprint arXiv:1506.02515*, 2015.

O. S. Lebedeva. Tensor conjugate-gradient-type method for Rayleigh quotient minimization in block QTT-format. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 26(5):465–489, 2011.

G. Lechuga, L. Le Brusquet, V. Perlbarg, L. Puybasset, D. Galanaud, and A. Tenenhaus. Discriminant analysis for multiway data. *Springer Proceedings in Mathematics and Statistics*, 2015.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010. URL http://www.ncbi.nlm.nih.gov/pubmed/20163403.

N. Lee and A. Cichocki. Estimating a few extreme singular values and vectors for large-scale matrices in Tensor Train format. *SIAM Journal on Matrix Analysis and Applications*, 36(3):994–1014, 2015.

N. Lee and A. Cichocki. Tensor train decompositions for higher-order regression with LASSO penalties. In *Workshop on Tensor Decompositions and Applications (TDA2016)*, 2016a.

N. Lee and A. Cichocki. Regularized computation of approximate pseudoinverse of large matrices using low-rank tensor train decompositions. *SIAM Journal on Matrix Analysis and Applications*, 37(2):598–623, 2016b. URL http://adsabs.harvard.edu/abs/2015arXiv150601959L.

Q. Li and D. Schonfeld. Multilinear discriminant analysis for higher-order tensor data classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(12):2524–2537, 2014.

X. Li, H. Zhou, and L. Li. Tucker tensor regression and neuroimaging analysis. *arXiv preprint arXiv:1304.5637*, 2013.

Z. Li, X. Liu, N. Xu, and J. Du. Experimental realization of a quantum support vector machine. *Physical Review Letters*, 114(14):140504, 2015.

S. Liao, T. Vejchodský, and R. Erban. Tensor methods for parameter estimation and bifurcation analysis of stochastic reaction networks. *Journal of the Royal Society Interface*, 12(108):20150233, 2015.

H. W. Lin and M. Tegmark. Why does deep and cheap learning work so well? *ArXiv e-prints*, 2016.

M. S. Litsarev and I. V. Oseledets. A low-rank approach to the computation of path integrals. *Journal of Computational Physics*, 305:557–574, 2016.

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*, 35(1):208–220, 2013.

C. Lubich, T. Rohwedder, R. Schneider, and B. Vandereycken. Dynamical approximation of hierarchical Tucker and tensor-train tensors. *SIAM Journal on Matrix Analysis and Applications*, 34(2):470–494, 2013.

C. Lubich, I. V. Oseledets, and B. Vandereycken. Time integration of tensor trains. *SIAM Journal on Numerical Analysis*, 53(2):917–941, 2015.

C. Lubich and I. V. Oseledets. A projector-splitting integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, 54(1):171–188, 2014. URL http://dx.doi.org/10.1007/s10543-013-0454-0.

L. Luo, Y. Xie, Z. Zhang, and W.-J. Li. Support matrix machines. In *The International Conference on Machine Learning (ICML)*, 2015a.

Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):3111–3124, 2015b.

T. Mach. Computing inner eigenvalues of matrices in tensor train matrix format. In *Numerical Mathematics and Advanced Applications 2011*, pages 781–788, 2013.

U. Manthe, H.-D. Meyer, and L. S. Cederbaum. Wave-packet dynamics within the multiconfiguration Hartree framework: General aspects and application to NOCl. *Journal of Chemical Physics*, 97:3199–3213, 1992.

H. Matsueda. Analytic optimization of a MERA network and its relevance to quantum integrability and wavelet. *arXiv preprint arXiv:1608.02205*, 2016.

H. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

P. J. Moreno, P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. *Advances in Neural Information Processing Systems*, 16:1385–1393, 2003.

T. D. Nguyen, T. Tran, D. Q. Phung, and S. Venkatesh. Tensor-variate Restricted Boltzmann Machines. In *AAAI*, pages 2887–2893, 2015.

J. Nocedal and S. J. Wright. *Sequential Quadratic Programming*. Springer, 2006.

A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 442–450, 2015.

A. Novikov, M. Trofimov, and I. V. Oseledets. Exponential machines. *arXiv preprint arXiv:1605.03795*, 2016.

J. Ogutu and H. Piepho. Regularized group regression methods for genomic prediction: Bridge, MCP, SCAD, Group Bridge, Group Lasso, Sparse Group Lasso, Group MCP and Group SCAD. In *BMC Proceedings*, volume 8, page S7. BioMed Central Ltd, 2014.

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

R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Physical Review B*, 78(15):155117, 2008.

I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011a.

I. V. Oseledets. Constructive representation of functions in low-rank tensor formats. *Constructive Approximation*, 37(1):1–18, 2012. URL http://dx.doi.org/10.1007/s00365-012-9175-x.

I. V. Oseledets and S. V. Dolgov. Solution of linear systems and matrix inversion in the TT-format. *SIAM Journal on Scientific Computing*, 34(5): A2718–A2739, 2012.

I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov. Tucker dimensionality reduction of three-dimensional arrays in linear time. *SIAM Journal on Matrix Analysis and Applications*, 30(3):939–956, 2008.

I. V. Oseledets, S. V. Dolgov, V. A. Kazeev, D. Savostyanov, O. Lebedeva, P. Zhlobich, T. Mach, and L. Song. TT-Toolbox, 2012. URL https://github.com/oseledets/TT-Toolbox.

S. Östlund and S. Rommer. Thermodynamic limit of density matrix renormalization. *Physical Review Letters*, 75(19):3537, 1995.

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*, 12(8):809–826, 2005.

A.-H. Phan, A. Cichocki, A. Uschmajew, P. Tichavsky, G. Luta, and D. Mandic. Tensor networks for latent variable analysis. Part I: Algorithms for tensor train decomposition. *ArXiv e-prints*, 2016.

A.-H. Phan, A. Cichocki, A. Uschmajew, P. Tichavský, G. Luta, and D. P. Mandic. Tensor networks for latent variable analysis. Part 2: Blind source separation and hamonic retrieval, *submitted to ArXiv e-prints*, 2017.

A. H. Phan and A. Cichocki. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear Theory and its Applications, IEICE*, 1(1):37–68, 2010.

A. H. Phan, P. Tichavský, and A. Cichocki. TENSORBOX: MATLAB package for tensor decomposition, 2012. URL http://www.bsp.brain.riken.jp/~phan/tensorbox.php.

H. N. Phien, H. D. Tuan, J. A. Bengua, and M. N. Do. Efficient tensor completion: Low-rank tensor train. *arXiv preprint arXiv:1601.01083*, 2016.

I. Pižorn and F. Verstraete. Variational numerical renormalization group: Bridging the gap between NRG and density matrix renormalization group. *Physical Review Letters*, 108:067202, 2012. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.067202.

T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. L. Liao. Why and when can deep–but not shallow–networks avoid the curse of dimensionality: A review. *arXiv preprint arXiv:1611.00740*, 2016.

G. Qi, Y. Sun, J. Gao, Y. Hu, and J. Li. Matrix variate RBM and its applications. *arXiv preprint arXiv:1601.00722*, 2016.

L. Qi. Hankel tensors: Associated Hankel matrices and Vandermonde decomposition. *Commun Math Sci*, 13(1):113–125, 2015.

G. Rabusseau and H. Kadri. Higher-order low-rank regression. *arXiv preprint arXiv:1602.06863*, 2016.

G. Raskutti and M. Yuan. Convex regularization for high-dimensional tensor regression. *arXiv preprint arXiv:1512.01215*, 2015.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*, volume 1. MIT Press, Cambridge, MA, 2006.

H. Rauhut, R. Schneider, and Z. Stojanac. Low rank tensor recovery via iterative hard thresholding. *arXiv preprint arXiv:1602.05217*, 2016.

P. Rebentrost, M. Mohseni, and S. Lloyd. Quantum support vector machine for big data classification. *Physical Review letters*, 113(13):130503, 2014.

J. Riihimäki, P. Jylänki, and A. Vehtari. Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *arXiv preprint arXiv:1207.3649*, 2012.

R. Rosipal and L. J. Trejo. Kernel partial least squares regression in reproducing kernel Hilbert space. *The Journal of Machine Learning Research*, 2:123, 2002.

R. Salakhutdinov and G. Hinton. Deep Boltzmann Machines. In *Proceedings of The 12th International Conference on Artificial Intelligence and Statistics (AISTATS'09)*, pages 448–445, 2009.

H. Sato, H. Kasai, and B. Mishra. Riemannian stochastic variance reduced gradient. *arXiv preprint arXiv:1702.05594*, 2017.

D. V. Savostyanov, S. V. Dolgov, J. M. Werner, and I. Kuprov. Exact NMR simulation of protein-size spin systems using tensor train formalism. *Physical Review B*, 90(8):085139, 2014.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

D. Schneider. Deeper and cheaper machine learning [top tech 2017]. *IEEE Spectrum*, 54(1):42–43, 2017.

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.

U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

U. Schollwöck. Matrix product state algorithms: DMRG, TEBD and relatives. In *Strongly Correlated Systems*, pages 67–98. Springer, 2013.

M. Schuld, I. Sinayskiy, and F. Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

T. J. Sejnowski. Higher-order Boltzmann machines. In *AIP Conference Proceedings 151 on Neural Networks for Computing*, pages 398–403, Woodbury, NY, USA, 1987. American Institute of Physics Inc. URL http://dl.acm.org/citation.cfm?id=24140.24200.

U. Shalit, D. Weinshall, and G. Chechik. Online learning in the manifold of low-rank matrices. In *Advances in Neural Information Processing Systems*, pages 2128–2136, 2010.

O. Sharir, R. Tamari, N. Cohen, and A. Shashua. Tensorial mixture models. *CoRR*, abs/1610.04167, 2016. URL http://arxiv.org/abs/1610.04167.

B. Sheehan and Y. Saad. Higher order orthogonal iteration of tensors (HOOI) and its relation to PCA and GLRAM. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 355–365. SIAM, 2007.

K. Shin, L. Sael, and U. Kang. Fully scalable methods for distributed tensor factorization. *IEEE Transactions on Knowledge and Data Engineering*, 29 (1):100–113, 2017.

M. Signoretto, L. De Lathauwer, and J. A. K. Suykens. A kernel-based framework to tensorial data analysis. *Neural Networks*, 24(8):861–874, 2011.

M. Signoretto, E. Olivetti, L. De Lathauwer, and J. A. K. Suykens. Classification of multichannel signals with cumulant-based kernels. *IEEE Transactions on Signal Processing*, 60(5):2304–2314, 2012.

A. Smola and V. Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161, 1997.

P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1*, pages 194–281. MIT Press, Cambridge, MA, 1986.

M. Steinlechner. *Riemannian Optimization for Solving High-Dimensional Problems with Low-Rank Tensor Structure*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2016a.

M. Steinlechner. Riemannian optimization for high-dimensional tensor completion. *SIAM Journal on Scientific Computing*, 38(5):S461–S484, 2016b.

E. M. Stoudenmire and D. J. Schwab. Supervised learning with quantum-inspired tensor networks. *arXiv preprint arXiv:1605.05775*, 2016. URL http://adsabs.harvard.edu/abs/2016arXiv160505775M.

E. M. Stoudenmire and S. R. White. ITensor Library Release v0.2.5. Technical report, Perimeter Institute for Theoretical Physics, May 2014. URL http://dx.doi.org/10.5281/zenodo.10068.

W. W. Sun and L. Li. Sparse low-rank tensor response regression. *arXiv preprint arXiv:1609.04523*, 2016.

Y. Sun, J. Gao, X. Hong, B. Mishra, and B. Yin. Heterogeneous tensor decomposition for clustering via manifold optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):476–489, 2016.

J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

M. Tan, I. Tsang, and L. Wang. Matching pursuit Lasso Part I and II: Sparse recovery over big dictionary. *IEEE Transactions on Signal Processing*, 63 (3):727–753, 2015.

J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. *SIAM Journal on Scientific Computing*, 35(5):S104–S125, 2013.

D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu. Supervised tensor learning. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 8–16. IEEE, 2005.

R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996a.

F. A. Tobar, S.-Y. Kung, and D. P Mandic. Multikernel least mean squares algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):265–277, 2014.

R. Tomioka and T. Suzuki. Convex tensor decomposition via structured Schatten norm regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1331–1339, 2013.

A. Uschmajew and B. Vandereycken. The geometry of algorithms using hierarchical tensors. *Linear Algebra and its Applications*, 439:133–166, 2013.

D. A. Vaccari. Taylorfit MPR. Simetrica, LLC., 2003. URL http://www.simetrica-llc.com/Products/MPR/index.html.

V. N. Vapnik and V. Vapnik. *Statistical Learning Theory*, volume 1. Wiley New York, 1998.

N. Vervliet and L. De Lathauwer. A randomized block sampling approach to Canonical Polyadic decomposition of large-scale tensors. *IEEE Transactions on Selected Topics Signal Processing*, 10(2):284–295, 2016. URL http://dx.doi.org/10.1109/JSTSP.2015.2503260.

N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. Tensorlab 3.0, Mar. 2016. URL http://www.tensorlab.net.

G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91(14):147902, 2003.

G. Vidal. Class of quantum many-body states that can be efficiently simulated. *Physical Review Letters*, 101(11):110501, 2008.

X. Wang and T. Xiang. Transfer-matrix density-matrix renormalization-group theory for thermodynamics of one-dimensional quantum systems. *Physical Review B*, 56(9):5061, 1997.

Y. Wang, H.-Y. Tung, A. Smola, and A. Anandkumar. Fast and guaranteed tensor decomposition via sketching. In *Advances in Neural Information Processing Systems*, pages 991–999, 2015.

G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar. Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Transaction on Graphics*, 31(4):80, 2012.

S. R. White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863, 1992.

S. R. White. Density matrix renormalization group algorithms with a single center site. *Physical Review B*, 72:180403, 2005.

K. Wimalawarne, M. Sugiyama, and R. Tomioka. Multitask learning meets tensor factorization: Task imputation via convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2825–2833, 2014.

K. Wimalawarne, R. Tomioka, and M. Sugiyama. Theoretical and experimental analyses of tensor-based regression and classification. *Neural Computation*, 28(4):686–715, 2016.

D. M. Witten. *A Penalized Matrix Decomposition, and its Applications*. PhD dissertation, Department of Statistics, Stanford University, 2010.

D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

T. Wu, A. R. Benson, and D. F. Gleich. General tensor spectral co-clustering for higher-order data. *arXiv preprint arXiv:1603.00395*, 2016.

T. Xiang and X. Wang. Density-matrix renormalization. *Lecture Notes in Physics*, 528, 1999.

Y. Yang and T. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016. URL `http://adsabs.harvard.edu/abs/2016arXiv160506391Y`.

A. Yeredor. Blind source separation via the second characteristic function. *Signal Processing*, 80(5):897–902, 2000.

T. Yokota, Q. Zhao, and A. Cichocki. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20): 5423–5436, 2016.

R. Yu, E. Y. Liu, and U. S. C. Edu. Learning from multiway data: Simple and efficient tensor regression. *Proceedings of the 33rd International Conference on Mahince Learning (ICML)*, 2016.

C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1582–1590, 2015a.

J. Zhang, Z. Wen, and Y. Zhang. Subspace methods with local refinements for eigenvalue computation using low-rank tensor-train format. *Journal of Scientific Computing*, pages 1–22, 2016.

Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel. Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(1):63–76, 2015b.

Q. Zhao, C. F. Caiafa, D. P. Mandic, L. Zhang, T. Ball, A. Schulze-Bonhage, and A. Cichocki. Multilinear subspace regression: An orthogonal tensor decomposition approach. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1269–1277. MIT, 2011.

Q. Zhao, C. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki. Higher order partial least squares (HOPLS): A generalized multilinear regression method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1660–1673, 2013a.

Q. Zhao, L. Zhang, and A. Cichocki. A tensor-variate Gaussian process for classification of multidimensional structured data. In *Proceedings of the Twenty-Seven AAAI Conference on Artificial Intelligence*, pages 1041–1047, 2013b.

Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki. Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data. *IEEE Signal Processing Magazine*, 30(4):137–148, 2013c.

Q. Zhao, G. Zhou, L. Zhang, and A. Cichocki. Tensor-variate Gaussian processes regression and its application to video surveillance. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1265–1269. IEEE, 2014.

Q. Zhao, L. Zhang, and A. Cichocki. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015.

Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S. I. Amari. Bayesian robust tensor factorization for incomplete multiway data. *IEEE Transactions on Neural Networks and Learning Systems*, 27(4):736–748, 2016.

X. B. Zhao, H. F. Shi, M. Lv, and L. Jing. Least squares twin support tensor machine for classification. *Journal of Information & Computational Science*, 11(12):4175–4189, 2014.

S. Zhe, Y. Qi, Y. Park, Z. Xu, I. Molloy, and S. Chari. DinTucker: Scaling up Gaussian process models on large multidimensional arrays. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016a. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11959/11888.

S. Zhe, P. Wang, K.-C. Lee, Z. Xu, J. Yang, Y. Park, and Y. Qi. Distributed flexible nonlinear tensor factorization. *arXiv preprint arXiv:1604.07928*, 2016b.

G. Zhou and A. Cichocki. TDALAB: Tensor Decomposition Laboratory. http://bsp.brain.riken.jp/TDALAB/, 2013. URL http://bsp.brain.riken.jp/TDALAB/.

H. Zhou, L. Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

T. Zhou, H. Qian, Z. Shen, and C. Xu. Riemannian tensor completion with side information. *arXiv preprint arXiv:1611.03993*, 2016.

D. Zwanziger. Fundamental modular region, Boltzmann factor and area law in lattice theory. *Nuclear Physics B*, 412(3):657–730, 1994.