KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Tensor-based independent component analysis: from instantaneous to convolutive mixtures

Frederik Van Eeghem

Supervisor:
Prof. dr. ir. L. De Lathauwer

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

January 2021

Tensor-based independent component analysis: from instantaneous to convolutive mixtures

Frederik VAN EEGHEM

Examination committee:
Prof. dr. ir.-arch. H. Neuckermans, chair
Prof. dr. ir. L. De Lathauwer, supervisor
Prof. dr. ir. S. Van Huffel
Prof. dr. ir. G. Samaey
Prof. dr. ir. I. Markovsky
(VUB, Belgium)
Prof. dr. ir. A.J. van der Veen
(TU Delft, The Netherlands)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

January 2021

© 2021 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Frederik Van Eeghem, Kasteelpark Arenberg 10 bus 2440, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

After all these years working towards this thesis, it is only fitting to thank the people that supported me along the way.

First of all, thank you Lieven for being my promotor and guiding me during my first steps in academia! It was inspiring to see your strive towards perfection in every aspect of research and teaching. I would also like to thank the chair and members of the examination committee who have taken the time to go through this thesis and providing me with valuable feedback and interesting questions.

Next, I would like to thank the colleagues in the office by whose side I have worked for several years. Martijn, Otto, Nico and Tom (who passed by often), thank you for the interesting discussions during our time together, both work- and non-work related. Working with you has been a pleasure and was a great influence at the start of my career. By extension, many thanks to the team members from Kortrijk (Ignat, Mikael, Michiel, Xiao-Feng, Geunseop) and the people from the biomed research group for our interesting talks and fun along the way.

Of course I have to thank my friends outside of work that have kept me sane during these years. Ward, Lucas, Els, thank you for the relaxing nights enjoying drinks in Leuven, as well as the cycling trips where we explored the wider region of Leuven. My friends from Ghent: Korneel, Gerald, Dries, Henri, Hans, Arne, Michiel, many thanks to you for the fun get-togethers throughout these years and the entertaining discussions we had. I am happy to see that we still meet often even though we live in different cities. During the past year, I also spent a lot of time in Kortrijk cycling, walking and playing sports in general with Thibault. Thank you for all the fun we had during these times and let's keep doing so!

Finally I would like to thank some people very close to me. My parents, Ma and Pa, thank you for supporting me and making me the person I am today! My older brother, Vincent, thank you for all the fun we had growing up and

the many talks we had on our future ambitions. Last but not least, I want to thank my girlfriend Anneleen; thank you for your support throughout these years working towards this thesis. I cannot imagine a life without you by my side!

Abstract

Data is everywhere, but it is only useful if information can be extracted from it. To this end, data analysis techniques have been developed to find features underlying the data. One class of data analysis techniques uses tensor decompositions. Tensors are higher-dimensional extensions of vectors and matrices and allow us to represent multiway data in a natural way. One of the key strengths of tensors is that their decompositions are unique under mild conditions without imposing additional constraints, unlike matrix decompositions. This property opens up many interesting applications of tensor decompositions, in particular with respect to blind source separation and independent component analysis.

Independent component analysis (ICA) tries to find the statistically independent signals underlying a mixture, which is useful in many fields such as telecommunications, speech separation and biomedical data analysis. These mixtures are often modeled as instantaneous mixtures of source signals and tensor methods that can blindly separate such mixtures are well known. However, in many applications a convolutive mixture model is more appropriate to take delays and reflections of the source signals into account. This is for instance the case for speech signals in a room or telecommunications signals impinging on antennas. When the mixtures are convolutive, additional structure arises in the separation problem. Current tensor-based methods in the literature do not fully exploit this Toeplitz or Hankel structure. This thesis presents new subspace-based methods that take more of this structure into account, which leads to efficient and more accurate results than the current state-of-the-art tensor methods. Additionally, relaxed uniqueness bounds are formulated that exploit the available structure as well. This thesis also presents a method for a related structured tensor decomposition in which all factor matrices have block-circulant structure.

Apart from exploiting the structure that appears in convolutive ICA, this thesis also presents how known techniques in instantaneous ICA can be ported to

the convolutive case. This includes coupled tensor decompositions to combine second- and fourth-order statistics, and the usage of incomplete tensors to reduce the computational complexity.

Another interesting question unrelated to convolutive ICA is whether we can compare underlying tensor factors without having to compute their full decompositions. This is highly relevant for any tensor classification problem but has only received limited attention in research. In this thesis, we present foundational theorems and algorithms that show how tensor factors can be compared in two modes for several underlying tensor decompositions. One of these methods is used further to develop an algorithm that is able to compare the cluster centers in two different datasets, without having to compute the actual clusters.

Beknopte samenvatting

Data is overal, maar is alleen nuttig als er informatie kan worden uitgehaald. Om dit te bereiken zijn data-analysetechnieken ontwikkeld die onderliggende kenmerken in data proberen te vinden. Eén type van data-analysetechnieken maakt gebruik van tensorontbindingen. Tensoren zijn meerdimensionale veralgemeningen van vectoren en matrices en laten toe om meerdimensionale data op een natuurlijke manier weer te geven. Eén van de belangrijkste voordelen van tensoren is dat hun ontbindingen uniek zijn onder milde voorwaarden. Bij tensorontbindingen hoeven er immers geen extra beperkingen opgelegd te worden om unieke ontbindingenen te verkrijgen, in tegenstelling tot matrixontbindingen. Deze eigenschap opent vele interessante toepassingen van tensorontbindingen, in het bijzonder met betrekking tot blinde bronscheiding en onafhankelijke componentenanalyse.

Onafhankelijke componentenanalyse (OCA) probeert de statistisch onafhankelijke signalen uit een mengsel terug te vinden, wat nuttig is in domeinen zoals telecommunicatie, spraakscheiding en biomedische data-analyse. Deze mengsels worden vaak gemodelleerd als ogenblikkelijke mengsels van bronsignalen en tensorgebaseerde algoritmes die dergelijke mengsels blind kunnen scheiden zijn bekend. In veel toepassingen is een convolutief mengmodel echter geschikter om ook eventuele vertragingen en reflecties van de bronsignalen in rekening te brengen. Dit is bijvoorbeeld het geval voor spraaksignalen in een ruimte of telecommunicatiesignalen die op antennes invallen. Bij dergelijke convolutieve mengsels ontstaat er extra structuur in het scheidingsprobleem. De huidige tensorgebaseerde methoden in de literatuur maken niet volledig gebruik van deze Toeplitz- of Hankelstructuur. Deze thesis presenteert nieuwe methoden gebaseerd op deelruimtes die meer rekening houden met deze structuur, wat leidt tot efficiëntere en nauwkeurigere resultaten dan de huidige state-of-the-art tensormethoden. Daarnaast worden er mildere uniciteitsvoorwaarden geformuleerd die eveneens de beschikbare structuur benutten. Deze thesis presenteert daarnaast ook een methode voor een gerelateerde gestructureerde tensorontbinding waarbij alle factormatrices een blok-circulante structuur

hebben.

Naast het uitbuiten van de structuur die voorkomt in convolutieve ICA, presenteert deze dissertatie ook hoe gekende technieken in ogenblikkelijke OCA kunnen worden geporteerd naar het convolutieve geval. Dit omvat gekoppelde tensorontbindingen om tweede- en vierdeordestatistieken te combineren, en het gebruik van onvolledige tensoren om de rekencomplexiteit te verminderen.

Een andere interessante vraag die geen verband houdt met convolutieve OCA is of onderliggende tensorfactoren kunnen vergeleken worden zonder hun volledige ontbindingen te moeten berekenen. Dit is zeer relevant voor elk tensorclassificatieprobleem, maar heeft tot dusver slechts beperkte aandacht gekregen in de onderzoekswereld. In deze thesis presenteren we fundamentele stellingen en algoritmen die laten zien hoe tensorfactoren in twee dimensies kunnen worden vergeleken voor verschillende onderliggende tensordecomposities. Een van deze methoden wordt vervolgens gebruikt om een algoritme te ontwikkelen dat in staat is om clustercentra van twee verschillende datasets te vergelijken zonder de eigenlijke clusters te hoeven berekenen.

List of Abbreviations

ALS	Alternating least squares
BSI	Blind system identification
BSS	Blind source separation
BTD	Block term decomposition
CCPD	Coupled canonical polyadic decompositions
CPD	Canonical polyadic decomposition
DFT	Discrete Fourier transform
DOA	Direction of arrival
GEVD	Generalized eigenvalue decomposition
HOS	Higher-order statistics
ICA	Independent component analysis
i.i.d.	Independent and identically distributed
JADE	Joint approximation diagonalization of eigenmatrices
LMLRA	Low multilinear rank approximation

MIMO	Multiple-input multiple-output
MISO	Multiple-input single-output
MLSVD	Multilinear singular value decomposition
MSE	Mean squared error
MRE	Mean relative error
NLS	Nonlinear least squares
PCA	Principal component analysis
REN	Relative error norm
RelErr	Relative error
SDF	Structured data fusion
SISO	Single-input single-output
SIMO	Single-input multiple-output
SNR	Signal-to-noise ratio
SOBI	Second-order blind identification
STFT	Short-time Fourier transform
SVD	Singular value decomposition
SVM	Support vector machines
ULA	Uniform linear array
ULRA	Unstructured low-rank approximation

Contents

Abstract iii

Beknopte samenvatting v

List of Abbreviations vii

Contents ix

List of Figures xv

List of Tables xxi

1 Introduction 1

1.1 Data and information 1

1.2 Tensors and their decompositions 2

1.3 Tensors for separating convolutive mixtures 4

1.4 Tensor similarity 6

1.5 Research aims 7

1.6 Outline of the thesis 8

1.6.1 Part 1: Structured tensor decompositions with focus on convolutive ICA and BSI 8

1.6.2 Part 2: Tensor similarity 10

2 Second-order tensor-based convolutive ICA: deconvolution versus tensorization 11

2.1 Introduction 13

2.2 Problem statement 14

2.3 Deconvolution after tensorization 16

2.4 Deconvolution before tensorization 18

2.5 Numerical experiments 20

2.5.1 Synthetic data 20

2.5.2 Application to rotating machines 22

2.6 Conclusion 24

3 Tensor decompositions with several block-Hankel factors and application in blind system identification 25

3.1 Introduction 26

3.1.1 Notation 27

3.1.2 Canonical Polyadic Decomposition 28

3.2 Blind System Identification 29

3.3 Identifiability condition for BSI 33

3.3.1 Deterministic uniqueness condition 34

3.3.2 Generic uniqueness results 37

3.4 Algorithms for BSI 38

3.4.1 Channel Subspace Method 40

3.4.2 Noise Subspace Method 42

3.5 Numerical experiments 44

3.5.1 Structured tensor decomposition 44

3.5.2 Application to blind system identification 46

3.6 Conclusions 53

4 Coupled and incomplete tensors in blind system identification 55

4.1 Introduction . . . 56

4.2 Canonical polyadic decomposition . . . 58

4.3 Problem statement . . . 59

4.3.1 Blind system identification . . . 59

4.3.2 Challenges . . . 61

4.4 Coupled decompositions . . . 63

4.4.1 Instantaneous mixture after deconvolution . . . 63

4.4.2 Convolutive mixtures . . . 64

4.4.3 Setting the weights . . . 67

4.5 Incomplete tensors . . . 68

4.6 Numerical experiments . . . 71

4.6.1 Choosing the weights for coupling . . . 71

4.6.2 Comparison of methods . . . 73

4.6.3 Incompleteness . . . 76

4.7 Conclusion . . . 77

5 Algorithms for canonical polyadic decomposition with block-circulant factors 79

5.1 Introduction . . . 80

5.2 Tensors and Problem statement . . . 81

5.2.1 Canonical polyadic decomposition . . . 81

5.2.2 Problem statement . . . 82

5.3 Algorithm and uniqueness . . . 82

5.4 Numerical experiments . . . 85

5.4.1 Circulant factors . . . 85

5.4.2 Block-circulant factors . . . 86

5.4.3 Parallel Wiener–Hammerstein system identification . . . 88

5.5 Conclusion 89

6 Tensor similarity in two modes **91**

6.1 Introduction 92

6.2 Tensor decompositions 94

6.2.1 Canonical polyadic decomposition 94

6.2.2 Block term decomposition in rank-$(L, L, 1)$ terms 96

6.2.3 Block term decomposition in rank-$(L, L, \cdot)$ terms 96

6.3 Problem statement and algorithm 97

6.3.1 Problem statement 97

6.3.2 Algorithm overview 98

6.3.3 Computational remarks 99

6.4 Full equality 102

6.4.1 CPD 102

6.4.2 BTD in rank-$(L, L, 1)$ terms 104

6.4.3 BTD in rank-$(L, L, \cdot)$ terms 105

6.5 Partial equality: the CPD case 106

6.5.1 Verifying partial similarity 106

6.5.2 Computing the common terms 109

6.5.3 Computing the distinct terms 110

6.6 Numerical experiments and applications 110

6.6.1 Influence of noise 110

6.6.2 Incorrectly estimated number of components 111

6.6.3 Application: Emitter movement detection 115

6.6.4 Application: Fluorescence spectrography 117

6.7 Conclusions 119

6.8 Proof of Theorem 4 120

6.9 Proof of Theorem 5 121

7 Determining whether two datasets cluster similarly without determining the clusters 123

7.1 Introduction 124

7.2 Problem statement 125

7.3 Tensors 126

7.3.1 Rank and Canonical Polyadic Decomposition 126

7.3.2 Tensor similarity 127

7.4 Comparing cluster centers 128

7.4.1 Step-by-step algorithm 128

7.4.2 Tensorization via Kronecker products 129

7.4.3 Tensorization via third-order moments 131

7.5 Numerical experiments 132

7.6 Conclusion 134

8 Conclusion 135

8.1 Contributions 135

8.2 Future perspectives 138

A Tensor similarity in chemometrics 141

A.1 Introduction 142

A.2 Tensors and matrices 144

A.3 Tensor decompositions 145

A.3.1 Multilinear singular value decomposition 145

A.3.2 Canonical polyadic decomposition 147

A.3.3 Block term decompositions 149

A.3.4 Block term decomposition in rank-$(L, L, 1)$ terms 149

A.3.5 Block term decomposition in rank-$(L, L, \cdot)$ terms 151
A.3.6 Block term decomposition in rank-(L, M, N) terms . . . 151
A.4 Tensor similarity 152
A.4.1 Problem statement 153
A.4.2 Tensor similarity without assumptions on decompositions 154
A.4.3 Similarity for specific decompositions 162
A.5 Similarity in chemometrics 169
A.5.1 Comparison of sets of mixtures without prior information 169
A.5.2 Comparison of analytes in mixtures (with prior information) 172
A.6 Conclusion 175

Bibliography **177**

Curriculum Vitae **193**

List of publications **195**

List of Figures

1.1 Visual representation of a vector (left), matrix (center) and third-order tensor (right). 2

1.2 Graphical illustration of a CPD. 4

1.3 Graphical illustration of several BTD variants. The top figure shows a BTD in multilinear rank-(L, M, N) terms, the middle figure a BTD in multilinear rank-$(L, L, 1)$ terms and the bottom figure a BTD in multilinear rank-$(L, L, \cdot)$ terms. 5

1.4 The thesis consists of two main parts: structured tensor decompositions (with focus on convolutive ICA / BSI) and tensor similarity 9

2.1 The unmixing step in convolutive ICA can be written as a block term decomposition in multilinear rank-$(L_{\text{tot}}, L_{\text{tot}}, \cdot)$ terms. . . 17

2.2 Using fewer observations enables faster deconvolution without sacrificing much accuracy for reasonable sample sizes. Here, the theoretical minimum number of observations needed is 11. 21

2.3 Using deconvolution or tensorization as the first step in the algorithm yields comparable results. The small difference between both approaches arises from the number of available observations. 23

2.4 Apart from a slight influence at 72 Hz, the fast deconvolution approach followed by SOBI cleanly separates a convolutive mixture of vibrational signals stemming from rotational machines. 24

3.1 Structure of the factor matrix $\mathbf{H}$. 31

3.2 Conceptual difference between a decomposition into rank-1 terms (top) and a decomposition into Hankel structured tensor terms (bottom). The hatched tensors have Hankel-structured factors. 34

3.3 Comparison of the median performance of the channel subspace method with a GEVD solution (—×—) or a NLS solution (—▲—) to (3.21), and the noise subspace method with a GEVD solution (- +-) or a NLS solution (- ⊖-) to (3.26) for a (noisy) tensor with two different banded block-Hankel factors. 46

3.4 Comparison of the mean performance for a blind system identification problem with $M = 5$, $R = 4$ and $L = 2$. The figure shows the channel subspace method with GEVD (- ⊖-) and with NLS (—●—), the noise subspace method with GEVD (- × -) and with NLS (—×—), the direct decomposition approach with GEVD (- - -) and with NLS (——), the approach from [88] using GEVD (- ▫ -) and using NLS (—▪—). 49

3.5 Comparison of the mean performance for a blind system identification problem with $M = 5$, $R = 4$ and $L = 2$. The figure shows the channel subspace method (- × -), noise subspace method (- +-) and optimization-based methods initialized in different ways. More specifically, a TENSORLAB implementation is shown with random initialization (—○—) and initialized with the noise subspace method (—●—), and the method from [59] is shown with random initialization (—△—) and initialized with the noise subspace method (—▲—). 51

3.6 Mean accuracy and time complexity of both the channel (blue) and noise (red) subspace methods when the maximum filter delay L is varied. 52

4.1 Block-Hankel structure of the factor matrix $\mathbf{H}$. This figure is adopted from [147] and slightly altered. 60

4.2 Graphical illustration of the two-step strategy using incomplete tensors. 70

4.3 The effect of different a relative weight ω_2 depends on the system and signal parameters. The graphs show the median over 1000 experiments at an SNR of 20 dB of a system with $L = 1$. . . . 72

4.4 Coupling second- and fourth-order information leads to better accuracies for methods starting with both deconvolution and tensorization. This improved accuracy comes at the price of a slower method. 74

4.5 Accuracy and time complexity comparison of methods using either only fourth-order information and methods combining fourth- and second-order information, this time for slightly longer filters ($L = 3$). 75

4.6 Using incomplete statistics leads to faster methods at the cost of a lower accuracy. The selection procedure of which entries to compute also impacts accuracy. The dashed black line denotes the time needed to construct the (incomplete) cumulant tensor. 77

5.1 Block-circulant structure of the factor matrix $\mathbf{U}^{(n)}$. 82

5.2 The method presented in this paper outperforms other methods, both as standalone method and as initialization for optimization-based methods. The figure shows the mean accuracy over 500 experiments. 87

5.3 Block scheme of a parallel Wiener–Hammerstein system. 88

6.1 Graphical illustration of a CPD. 95

6.2 Graphical illustration of a BTD in multilinear rank-$(L, L, 1)$ terms. 96

6.3 Graphical illustration of a BTD in multilinear rank-$(L, L, \cdot)$ terms. 97

6.4 The goal of the paper is to verify whether the factor matrices of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ in the first two modes are equal without explicitly computing the decompositions and using only linear algebra, illustrated here for the case in which both tensors admit a CPD. 98

6.5 Effects on the principal angles of varying noise levels on tensors that are essentially equal in two modes (a) and not essentially equal in two modes (b). For visual clarity, the largest principal value is shown in red. 112

6.6 The relative errors on the first two factor matrices when the CPDs are explicitly computed show similar behavior as the principal angles. Note that the errors on both factors almost coincide. . 113

6.7 The similarity method is orders of magnitude faster than the method computing the CPDs and is independent of the noise level. 113

6.8 Effects on the principal angles of an incorrectly estimated number of components, which is either chosen too low (a) or too high (b) for tensors that are essentially equal in two modes. For visual clarity, the largest principal value is shown in red. 114

6.9 Direction of arrival estimation tries to retrieve the values α_r using solely the information obtained at the antenna array. . . 115

6.10 Using essential equality in two modes one can detect whether any emitter has displaced, which leads to lighter computational loads when there is no displacement. 117

6.11 The first 3 singular values of $\mathbf{T}^{(1)}_{10100\times 61}$ and $\mathbf{T}^{(2)}_{10100\times 61}$ contain most of the information. 118

7.1 Effects of various parameters on the principal angles when the cluster centers of both datasets are the same. 133

A.1 Graphical illustration of a MLSVD. 146

A.2 Illustration of the MLSVD of a tensor with low multilinear rank (assuming the blue core tensor has full multilinear rank). The blue parts of the factor matrices denote the vectors that span the mode-n spaces of $\mathcal{T}$. 146

A.3 Graphical illustration of a CPD. 148

A.4 Graphical illustration of a BTD in multilinear rank-$(L, L, 1)$ terms. 150

A.5 Graphical illustration of a BTD in multilinear rank-$(L, L, \cdot)$ terms. 152

A.6 Graphical illustration of a BTD in multilinear rank-(L, M, N) terms. 152

A.7 Graphical representation of comparing the factor matrices of $\mathcal{A}$ and $\mathcal{B}$ to investigate essential equality in the first two modes for the case in which both tensors admit a CPD. (Adopted with changes from [145]) 154

A.8 The first three singular values of the matrix unfoldings $\mathbf{A}_{[2,3;1]}$, $\mathbf{A}_{[1,3;2]}$ and $\mathbf{A}_{[1,2;3]}$ contain most of the information. 172

A.9 The first three singular values of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$ contain most of the information. Adopted with changes from [145]. 174

List of Tables

3.1 For a given pair (M, L) the entry corresponds to the maximal value of R for which conditions (3.10a)–(3.10c) hold. 38

3.2 Maximum value for R for which the conditions from [164, 131] and Proposition 3.3.2 generically guarantee uniqueness of the channel parameter matrix $\mathbf{P}$ in (3.4). 39

3.3 Average computation time over 200 experiments for various BSI methods computed on a standard laptop. The considered system has parameters $M = 5$, $R = 4$ and $L = 2$ and outputs of length 10^5 with 15 dB SNR. The table only shows the decomposition time, i.e., the time needed to compute the higher-order statistics is not included. 49

4.1 Classification of cumulant entries based on the repeated indices of the spatial parameters. 76

5.1 Our method outperforms a naive implementation of [63] and is nearly optimal when compared to a full optimization-based solution. The table shows the mean estimation error over 100 experiments expressed in dB for various SNRs. 86

5.2 Execution times of various algorithms. 87

6.1 The principal angles clearly indicate which mixtures consist of the same components. 119

A.1 Concentrations of the analytes Tyrosine, Tryptophan and Phenylalanine in each mixture, represented by the frontal slices of $\mathcal{A}$ and $\mathcal{B}$ which contain the excitation-emission spectra of the mixtures. 170

A.2 Largest principal angles for the comparison of the spaces spanned by the matrix unfoldings of $\mathcal{A}$ and $\mathcal{B}$ in each mode. 171

A.3 Largest principal angles for the comparison of the spaces spanned by the matrix unfoldings of $\mathcal{A}$ and $\mathcal{B}$ in each mode. 171

A.4 The principal angles clearly indicate which mixtures consist of the same components. Adopted from [145]. 175

Chapter 1

Introduction

1.1 Data and information

Data is everywhere and is generated at an increasingly fast pace. For example, the oil and gas industry equips installations with sensors to obtain data on machine vibrations, temperatures, pressure, proximity and more. Autonomous vehicles, often equipped with cameras, Lidar and GPS, generate from 4 to 10 terabytes of data per hour, depending on the number of sensors and their resolution [6]. Another example can be found in social media, where 4 petabytes of new data is generated per day on Facebook alone [160].

The real value of all this data is the information and insights that can be extracted from it. Sensor data of machine vibrations can for instance be used to predict when maintenance is due. Data captured by autonomous vehicles gives a view on the surrounding area and possible dangers, which can in turn be used to safely steer the car to its destination. Social media data gives insights into people's interests and relations, which is extremely valuable information for targeted advertisements.

Extracting information from data can be done in many ways. Possible methods include supervised machine learning techniques, clustering, matrix or tensor factorizations and statistical modeling. The most suitable method for a specific application strongly depends on the application at hand, the available data and prior information. Google's "AI First" strategy illustrates that the focus of many research groups and companies is to improve these methods that are able to find patterns, structure and information in the collected data. Within this broader context, the focus of this thesis lies in improving structured tensor



Figure 1.1: Visual representation of a vector (left), matrix (center) and third-order tensor (right).

decompositions and developing tensor similarity, which both allow to extract information from tensor data.

1.2 Tensors and their decompositions

Tensors are third- or higher-order generalizations of vectors and matrices and are illustrated graphically in Figure 1.1. Tensors and tensor-based analysis have gained popularity in the past decades, which can largely be attributed to three reasons.

First, tensors allow a natural representation of multi-way data. For instance, monitoring the temperature in a room leads to a four-dimensional tensor with one temporal dimension and three spacial dimensions. Another example can be found on growth data of children, where the first mode contains the subjects, the second mode contains the measurement variables (height, weight, head circumference, ...) and the third mode contains the age. When all measurement variables are registered for each subject at different ages, a third-order tensor is obtained [83]. Many more examples of applications where tensors appear in a natural way can be found in domains such as psychometrics, chemometrics, signal processing, numerical linear algebra and more [25, 64, 81, 110, 112]. Though it is possible to represent multiway data in a matrix or even in a vector, it requires an artificial combination of dimensions and obscures the relations between different dimensions.

A second strength of tensors is that they can alleviate the curse of dimensionality via low-rank representations. The curse of dimensionality refers to the exponential increase of required data when increasing the dimensionality. For instance, a square order-N tensor with size 10 in each mode has 10^N entries, which rapidly leads to large-scale datasets for higher dimensionalities. However, many of these large-scale datasets actually have a low-rank underlying structure and can be efficiently represented or approximated by tensor decompositions such as multilinear singular value decompositions, canonical polyadic decompositions,

tensor trains or hierarchical Tucker decompositions [64, 100]. By working directly with the tensor decompositions rather than with the full tensors, one can still efficiently work on large-scale problems. Note that certain approaches exploit incompleteness or sparseness of tensors to further reduce the computational complexity in large datasets [155, 154].

The third strength of tensors lies in the uniqueness of tensor decompositions under mild conditions. Matrix decompositions are generally not unique without imposing additional constraints. Consider a matrix $\mathbf{M} \in \mathbb{C}^{I \times J}$ that has rank $R \leq \min\{I, J\}$. This matrix can be decomposed as

$$\mathbf{M} = \mathbf{A}\mathbf{B}^{\mathrm{T}},$$

in which $\mathbf{A} \in \mathbb{C}^{I \times R}$ and $\mathbf{B} \in \mathbb{C}^{J \times R}$ have full column rank. In this decomposition, it is possible to post- and premultiply the factors with a nonsingular matrix and its inverse without changing the outcome, e.g.:

$$\mathbf{M} = (\mathbf{A}\mathbf{G}^{-1})(\mathbf{G}\mathbf{B}^{\mathrm{T}}) = \tilde{\mathbf{A}}\tilde{\mathbf{B}}^{\mathrm{T}},$$

with $\mathbf{G} \in \mathbb{C}^{R \times R}$ a nonsingular matrix. This immediately illustrates that matrix decompositions are not unique. The only way to obtain unique matrix decompositions is by imposing additional constraints on the factors, such as orthogonality in the singular value decomposition. In the case of many tensor decompositions, no additional constraints are needed to obtain uniqueness. This property is important because it implies that the different terms of a decomposition will not change every time the decomposition is recomputed, which in turn support interpretation of the terms.

The remainder of this section will introduce two types of tensor decompositions, the canonical polyadic decomposition and the block-term decomposition, that will play an important role throughout this thesis.

Before the canonical polyadic decomposition (CPD) can be defined, the concept of a rank-1 tensor must be introduced. A rank-1 term $\mathcal{A}$ of order N is defined as a tensor that can be written as the outer product of N nonzero vectors. The CPD then decomposes a tensor into a linear combination of a minimal number of rank-1 terms, as illustrated graphically in Figure 1.2. While rank-1 decompositions of matrices are not unique without additional constraints, it has been shown that a tensor CPD is essentially unique under mild conditions [84, 108, 49, 50, 52]. This means that, under mild conditions, a CPD is unique up to permutation of its terms and possible scaling and counterscaling of the factor vectors within each term. This property makes CPDs powerful tools in signal processing, and especially in signal separation applications [81, 110, 29, 15]. For instance, the CPD can be used to separate the heartbeats of a fetus and mother

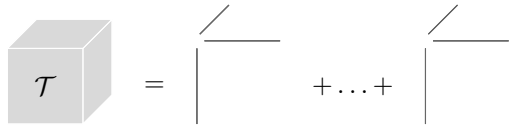


Figure 1.2: Graphical illustration of a CPD.

from an ECG [36]. Algorithms to compute such CPDs have been extensively researched [81, 30, 51, 52, 155, 150].

In many applications, the rank-1 constraint on the terms of the CPD is too strict. In these cases, block-term decompositions can be used to model more complex phenomena [31]. In chemometrics for instance, a tensorization technique called Löwnerization allows to construct a tensor from just one excitation-emission matrix stemming from a mixture of amino acids rather than stacking the excitation-emission matrices of several mixtures. In this new approach, each of the amino acids contributes one block-term to the tensor instead of the rank-1 contributions typically seen in chemometrics. This means that the different mixture components can be identified by computing a BTD of the constructed tensor [46, 15]. The most general variant of a block-term decomposition allows low-rank contributions in each mode by decomposing a tensor into multilinear rank (L, M, N)-terms as illustrated graphically in Figure 1.3. Other variants include the BTD in multilinear rank-$(L, L, 1)$, where low-rank contributions are only allowed in the first two modes, and the BTD in multilinear rank-$(L, L, \cdot)$ terms, which allows full rank contributions in the third mode [31]. It has been shown that these block-term decomposition are unique up to trivial indeterminacies under mild conditions as well [31, 54, 55, 32, 117]. Algorithms for computing these BTDs can be found in e.g. [41, 113, 54, 55, 1, 16, 114].

1.3 Tensors for separating convolutive mixtures

Many signals in practice can be modeled as a mixture of unknown components that are of interest. One example can be found in biomedical data processing: fetal ECG measurements include the heartbeats of both mother and child [36]. Similarly, EEG measurements are the sum of the actual signals of interest and artifacts due to eye movements and muscle contractions [71, 73]. A similar situation can be found in telecommunications: if several signals impinge on an antenna array, we receive a mixture while we wish to obtain the original signals [2].

In these examples, the challenge is that the underlying signals of interest

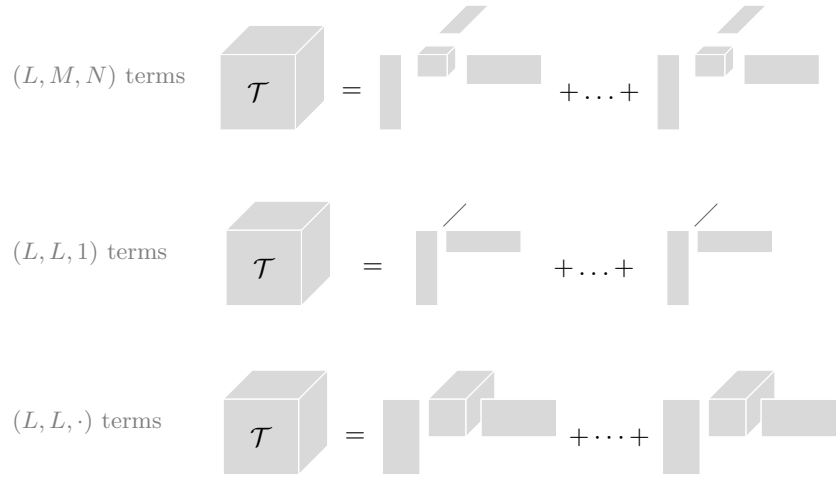


Figure 1.3: Graphical illustration of several BTD variants. The top figure shows a BTD in multilinear rank-(L, M, N) terms, the middle figure a BTD in multilinear rank-$(L, L, 1)$ terms and the bottom figure a BTD in multilinear rank-$(L, L, \cdot)$ terms.

are hard or even impossible to measure. We cannot isolate the fetal heartbeat in measurements, the mother's heartbeat will always be there. In telecommunications, we could send training sequences to identify the mixing coefficients, but these sequences take up a significant part of the available bandwidth. In these types of applications, blind source separation (BSS) is an attractive solution. BSS separates the original sources making up a mixture without prior knowledge of these sources or the mixing coefficient. Sadly, this problem cannot be solved without imposing additional constraints on either the sources or the mixing vectors [132]. Depending on the application, popular assumptions are statistically independent inputs [28], finite alphabet [139], constant modulus [133, 137] or low-rank mixing vectors [14]. This thesis zooms in on blind source separation assuming statistically independent inputs, which is sometimes referred to as independent component analysis (ICA) since it tries to recover the underlying independent components. A typical example of mixtures of independent components can be found in the cocktail party problem, where the goal is to retrieve individual speakers from audio recordings of a cocktail party.

The most simplified way to solve the cocktail party problem is to assume that the audio signals of the individual speakers do not reflect on walls or have any time delays before reaching the microphones. This implies that the

recordings can be seen as instantaneous mixtures of the individual speakers. Under this simplification, tensor-based methods have been extensively studied. One approach uses fourth-order statistics of the mixture signals to construct a fourth-order tensor which, under the assumption of independent source signals, admits a CPD from which the mixing coefficients can easily be extracted [22, 34, 37, 28]. Another approach stacks second-order statistics of the mixture signals to obtain a third-order tensor, which again admits a CPD under the independent source assumption from which the mixture coefficients can be retrieved [10].

In a more realistic setting, the individual speech signals in the cocktail party problem will reflect on the walls and will have time delays before reaching the microphones [2]. A similar scenario can be found in telecommunications, where signals impinging on an antenna may reflect on buildings as well. In these scenarios, convolutive mixture models are more fitting than simple instantaneous mixture models. It has been shown in the literature that tensor-based problems are still useful tools to separate these convolutive mixtures, but the convolution introduces additional structure in the problem [59, 13]. In most cases, this structure takes the form of Toeplitz- or Hankel-structured factor matrices [23, 59]. Whereas tensor-based methods for separating instantaneous mixtures are well understood, most research for convolutive mixtures of independent components has been focused on other approaches [103] and only a few tensor-based approaches have been published [164, 59, 13, 88]. Note that in this context, the terms convolutive ICA and blind system identification (BSI) are used interchangeably. The tensor-based methods that do exist in the literature do not fully exploit the available structure in the tensor decompositions, which leads to suboptimal accuracy results and to rather strict uniqueness bounds. One of the key topics in this thesis concerns capitalizing on this structure to improve on these existing methods. Additionally, techniques such as coupled tensor decompositions and incomplete tensors are considered to further improve on the state-of-the-art in tensor-based blind system identification.

1.4 Tensor similarity

A large part of the current tensor research aims to find underlying factors through various decompositions [31, 81, 25, 110]. However, knowledge of the exact underlying factors is not always needed in signal processing and machine learning. In classification problems, the "distance from" known data points is more important than the actual absolute values. Consider for example emitter localization, where the goal is to determine the position of an emitter using an antenna array. This problem can be written as a tensor decomposition [109, 14].

To determine whether the emitter has moved, the location information from two different tensors must be compared, which resides in the tensor factors. One naive approach is to fully compute the position on both time instances and compare the positions afterwards, but this is computationally intensive. It has been shown that it is possible to compare underlying tensor factors without explicitly computing them [33], which is not possible in the matrix case. This thesis will develop new results in tensor similarity that allow to compare tensor factors in two modes for CPDs and various block term decompositions.

1.5 Research aims

The research presented in this thesis is centered around following four key aims:

1. *Improving the performance of tensor-based convolutive ICA methods.*

   Performance improvements in convolutive ICA or blind system identification are made in several ways. First, the accuracy of tensor-based methods for BSI is improved by either capitalizing on the available structure in the tensor factors or by combining more information via coupled tensor decompositions. Second, computations are sped up where possible by developed algebraic methods for BSI. Third, the uniqueness conditions for tensor-based BSI are further relaxed.

2. *Exploiting structure in tensor decompositions.*

   Results on exploitation of structure in tensor decomposition can already be found in the literature and has proven to significantly improve accuracy and computation time of these decompositions [116, 115]. We aim to expand on this, both for the specific structure that arises in tensor-based BSI as for other cases.

3. *Porting techniques from instantaneous mixtures to the convolutive case.*

   As stated in the introduction, instantaneous ICA has been well studied for years and several techniques to improve on existing methods have been developed. We aim to port some of the knowledge of instantaneous techniques to the convolutive case to obtain better algorithms. Examples of this are coupled tensor decompositions that use both second- and fourth-order information and incomplete tensor decompositions.

4. *Comparing underlying factors of tensors.*

   A large part of tensor research is focused on finding the factors of tensor decompositions. However, in some applications it suffices to compare

the factors of two tensors rather than explicitly computing them. Very few results on this have been published in the literature. We aim to further develop this field by finding tensor similarity methods that allow to compare tensor factors in two modes in a time-efficient way.

1.6 Outline of the thesis

This section gives an overview of the chapters in this thesis and how they are interlinked. The main structure within this thesis is visualized in Figure 1.4 and consist of two parts. The first chapters discuss convolutive ICA and exploiting structure, whereas the last chapters focus on the comparison of factors and tensor similarity. Each chapter of the thesis is a standalone paper that can be read separately from the rest of the text. In Chapter 8, the thesis results are summarized and possible next steps are outlined.

1.6.1 Part 1: Structured tensor decompositions with focus on convolutive ICA and BSI

The first part of the thesis focuses on extracting information from data, more specifically by computing structured tensor decompositions. By exploiting structure that is available, or by combining existing methods in novel settings, the goal of this part is to improve on the state-of-the-art methods for structured tensor decompositions, both in terms of accuracy and computation time. Blind system identification of FIR systems is used throughout this part as a motivating example, explaining the focus on convolutions and (circulant) Toeplitz structure.

Chapter 2 contains a study on the exploitation of structure in tensor-based convolutive ICA using second-order statistics. We show that any tensor-based method for convolutive ICA consists of two steps: a tensorization step and a deconvolution step. We investigate in which order these two steps should be done for optimal results. On top of this, we develop a faster deconvolution approach that uses only partial information.

Chapter 3 then looks at tensor-based methods using fourth-order statistics in blind system identification (BSI). BSI is an analogous to convolutive ICA, but instead of aiming to retrieve the statistically independent components, we wish to retrieve the mixing coefficients. Tensor-based methods for BSI exist, but they do not exploit the available block-Hankel structure that arises in the factor matrices. This chapter devises an entirely new subspace-based algorithm that exploits the structure in two dimensions. On top of

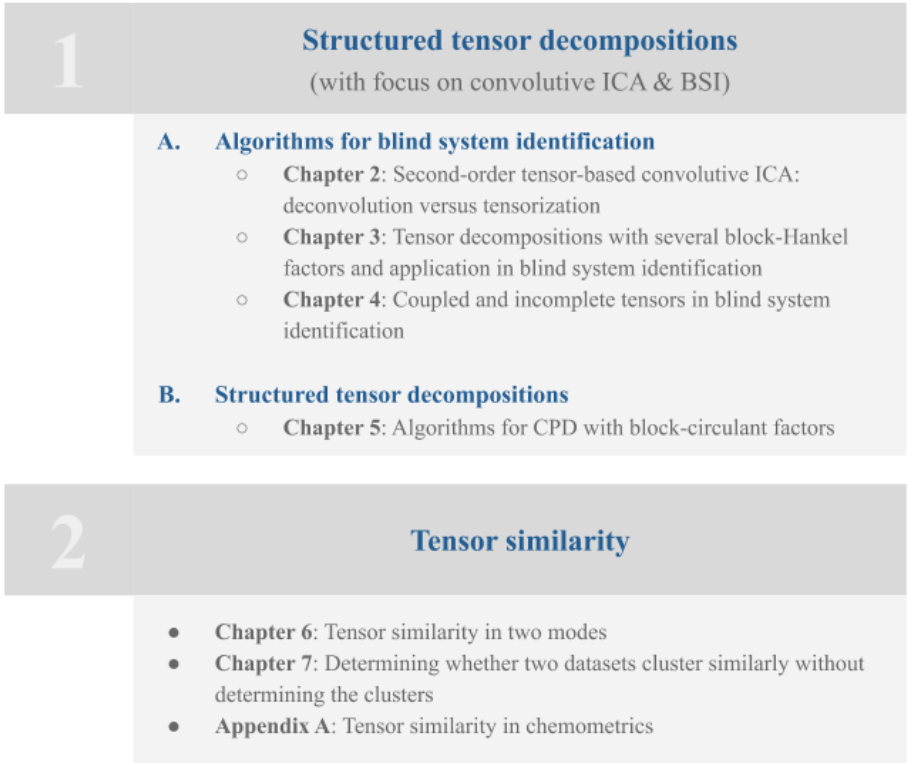


Figure 1.4: The thesis consists of two main parts: structured tensor decompositions (with focus on convolutive ICA / BSI) and tensor similarity

this, new uniqueness bounds are derived that are much more relaxed that the existing ones. In brief, these uniqueness bounds theoretically prove the maximum number of independent components that can be recovered from a set of mixtures. We show that this new algorithm provides fast and accurate results while relaxing theoretical bounds.

Chapter 4 takes the results from the previous chapters and the literature and further improves methods for BSI using two strategies that can be ported from the field of instantaneous mixture separation. The first strategy improves the accuracy by combing second- and higher-order information through coupled tensor decompositions. The second approach uses incomplete tensors to reduce the time complexity of the method with little impact on the accuracy.

Chapter 5 strays further from the structure that arises in BSI applications,

but looks at a specific type of structured tensor decompositions. More specifically, the structure of CPD decompositions with all block-circulant factors are exploited in a new method that transforms the entire data tensor to the frequency domain and applies element-wise divisions. This yields a tensor of lower rank with no additional structure, which is far easier to decompose.

1.6.2 Part 2: Tensor similarity

Where the focus of the first part lies in extracting underlying variables from data, this second part focuses of comparing tensors with each other. By defining similarity measures and algorithms, the comparison of higher-order tensors can be reduced to a small set angles that capture the similarity them. This offers new tools and possibilities in any setting where tensors need to be compared rather than decomposed, as is often the case in classification problems.

Chapter 6 focuses on tensor similarity in two modes. In some applications we wish to compare tensor factors rather than explicitly compute them. Though it is possible to decompose both tensors and compare their factors afterwards, more efficient methods can be found. This chapter presents a fast similarity method that indicates whether the factors in two modes are equal op to scaling and permutation without explicitly computing them. Equality conditions are derived that ensure the theoretical validity of the approach.

Chapter 7 illustrates the strength of the previous chapter by applying the ideas to clustering. We assess whether two datasets consist of clusters with the same centers or not. This is valuable information if you wish to merge consumer datasets of different countries for instance. By tensorizing both datasets, tensor similarity can be used to compare the cluster centers without having to explicitly determine the clusters.

Appendix A gives a more accessible overview of recent advances in tensor similarity tailored to the chemometrics community. This includes part of the results from Chapter 5 but presented in a way that allows researchers in chemometrics without a deep mathematical background to use the developed methods in practice.

Chapter 2

Second-order tensor-based convolutive ICA: deconvolution versus tensorization

Abstract — Independent component analysis (ICA) research has been driven by various applications in biomedical signal separation, telecommunications, speech analysis, and more. One particular class of algorithms for instantaneous ICA uses tensors, which have useful properties. In an attempt to port these properties to convolutive methods, we zoom in on an existing method that uses second-order statistics. By pointing out links in the literature, we show that this method is in fact a typical tensor-based method, even though this was not recognized by the authors at the time.

The existing method mentioned above can be interpreted as a tensorization step followed by a deconvolution step. However, as sometimes done in literature, one may consider using the opposite approach; starting with a deconvolution step and then tensorizing the remaining instantaneous mixture. Because subspace-based deconvolution can be slow, we propose a fast variant which uses only partial information. We then use this variant to compare the approach starting with tensorization and the one starting with deconvolution.

Reference — This chapter is a slightly adapted version of [141]. The candidate did the research and wrote the article under the guidance of the coauthor.

2.1 Introduction

Various signals in biomedical data processing and array processing can be interpreted as an instantaneous mixture of statistically independent components. To retrieve these components, one can turn to independent component analysis (ICA) [28]. However, in several applications, the underlying independent components are not mixed instantaneously because of path length differences and reflections. Among others, this is the case for speech signals, telecommunications, and seismic signals [2]. For these applications, a convolutive mixture model is more appropriate.

Tensors are higher-dimensional generalizations of vectors and matrices. A key strength of tensors is the uniqueness of their decompositions under mild conditions [50, 31]. Moreover, several decompositions can be computed algebraically [51]. These properties make tensors useful tools in chemometrics, psychometrics, and signal separation [25, 81, 110].

For both instantaneous as convolutive ICA, a variety of algorithms have been developed [28, 71]. Of particular interest are tensor-based methods, which are well-established for instantaneous ICA [10, 22, 28]. In an attempt to port the favorable tensor properties to convolutive ICA, several tensor-based algorithms have been presented for convolutive mixtures [58, 13]. In these methods, the convolution gives rise to additional structure, which is often not exploited. Since taking this structure into account may lead to faster or more accurate methods, recent advances focus on exploiting the structure [147]. In this paper, we focus on convolutive mixtures of temporally coherent and mutually independent components. A separation algorithm for this model using second-order statistics was coined in [13], and can actually be seen as the convolutive extension of the popular SOBI algorithm for instantaneous ICA [10]. Though not recognized by the authors, the method in [13] is in fact a tensor-based approach. As a first contribution, we will clarify the link between the method from [13] and tensor decompositions.

Most convolutive ICA methods involve a deconvolution step and a separation step. The method from [13] first separates the mixture and subsequently deconvolves the resulting filtered versions of the original sources. The opposite approach, in which the data are deconvolved first, is quite popular because it allows one to use any instantaneous ICA algorithm after the deconvolution. Deconvolving is typically done using second-order statistics [61, 94] or subspace-based techniques [91, 138, 139]. Subspace-based techniques avoid computing any statistics, but they involve large-size matrices and may be slow [93]. For single-input multiple-output (SIMO) systems, this has been countered by using only part of the data [90]. As a second contribution, we extend this idea to MIMO

systems to obtain faster deconvolution. We give a lower bound for the number of needed observations and provide a relative measure to determine reasonable values for the number of observations in practice. Numerical experiments support our analysis.

Notations: scalars are represented by normal lowercase letters (e.g., a), vectors by bold lowercase letters (e.g. $\mathbf{a}$), matrices by bold uppercase letters (e.g. $\mathbf{A}$) and tensors by calligraphic letters (e.g. $\mathcal{A}$). The Kronecker product is denoted by $\otimes$. The transpose and conjugate transpose are represented by $\cdot^{\mathrm{T}}$ and $\cdot^{\mathrm{H}}$, respectively. Commas and semicolons are used for horizontal and vertical concatenation, respectively. Bars are used to represent dropped block-rows. For instance, consider $\mathbf{A} = \left[\mathbf{A}^{(1)}; \cdots ; \mathbf{A}^{(N)}\right]$, then $\overline{\mathbf{A}} = \left[\mathbf{A}^{(2)}; \cdots ; \mathbf{A}^{(N)}\right]$ and $\underline{\mathbf{A}} = \left[\mathbf{A}^{(1)}; \cdots ; \mathbf{A}^{(N-1)}\right]$. The mathematical expectation is denoted by $\mathbb{E}\{\cdot\}$.

2.2 Problem statement

A convolutive mixture with M outputs $x_m(t)$ and R inputs $s_r(t)$ can be written as

$$x_m(t) = \sum_{r=1}^{R} \sum_{l=0}^{L} h_{mr}(l) s_r(t-l) + v(t) \qquad \text{for } m \in \{1, \ldots, M\}. \tag{2.1}$$

In this equation, $h_{mr}(l)$ with $l \in \{0, \ldots, L\}$ represents the filter between the rth input and mth output. The maximum filter delay is represented by L. The additive noise is represented by $v(t)$. Note that in this chapter, and in the rest of the thesis, the focus lies on finite impulse response (FIR) systems. This means that the results are not directly applicable on infinite impulse response systems (IIR). However, one can use a FIR model to approximate an IIR model, which may be a reasonable approximation if the norms of the IIR filter coefficients decay rapidly.

If the system is strictly overdetermined, i.e., if there are strictly more outputs than inputs, equation (2.1) can be written as an overdetermined matrix equation

$$\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{V}, \tag{2.2}$$

in which $\mathbf{X} \in \mathbb{C}^{ML' \times N}$ represents the outputs, $\mathbf{S} \in \mathbb{C}^{RL_{\mathrm{tot}} \times N}$ contains (lagged versions of) the inputs and $\mathbf{V} \in \mathbb{C}^{ML' \times N}$ contains the additive noise. The value L' represents the number of lagged outputs taken into account and is chosen such that the mixing matrix $\mathbf{H} \in \mathbb{C}^{ML' \times RL_{\mathrm{tot}}}$ is tall, i.e., such that $ML' \geq RL_{\mathrm{tot}}$, with $L_{\mathrm{tot}} = L + L'$. Note that including lagged versions of the

outputs to obtain a tall matrix $\mathbf{H}$, sometimes called temporal smoothing, is only possible because the system is strictly overdetermined.

The columns $\mathbf{x}(n)$ and $\mathbf{s}(n)$ of the matrices $\mathbf{X}$ and $\mathbf{S}$ take the form

$$\begin{aligned}\mathbf{x}(n) = &[x_1(n), \ldots, x_M(n), x_1(n-1), \ldots,\\ &x_M(n-1), \ldots, x_M(n-L'+1)]^{\mathrm{T}},\\ \mathbf{s}(n) = &[s_1(n), s_1(n-1), \ldots, s_1(n-L_{\mathrm{tot}}+1), \ldots,\\ &s_R(n), \ldots, s_R(n-L_{\mathrm{tot}}+1)]^{\mathrm{T}}. \end{aligned} \tag{2.3}$$

Equation (2.3) shows that if the rows of $\mathbf{S}$ are properly permuted, it has block-Toeplitz structure for $L > 0$. Denote the row-permuted version of $\mathbf{S}$ having block-Toeplitz structure by $\widetilde{\mathbf{S}} = \left[\widetilde{\mathbf{S}}^{(1)}; \ldots; \widetilde{\mathbf{S}}^{(L_{\mathrm{tot}})}\right] \in \mathbb{C}^{RL_{\mathrm{tot}} \times N}$, in which the $\widetilde{\mathbf{S}}^{(l)} \in \mathbb{C}^{R \times N}$ are shifted versions of each other.

The mixing matrix $\mathbf{H} \in \mathbb{C}^{ML' \times RL_{\mathrm{tot}}}$ from (2.2) is given by

$$\mathbf{H} = [\mathbf{H}_1 \quad \mathbf{H}_2 \quad \cdots \quad \mathbf{H}_R],$$

in which

$$\mathbf{h}_q(l) = \begin{bmatrix} h_{1q}(l) & h_{2q}(l) & \cdots & h_{Mq}(l) \end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^M,$$

and

$$\mathbf{H}_q = \begin{bmatrix} \mathbf{h}_q(0) & \cdots & \mathbf{h}_q(L) & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & \mathbf{h}_q(0) & \cdots & \mathbf{h}_q(L) \end{bmatrix} \in \mathbb{C}^{ML' \times L_{\mathrm{tot}}}.$$

Finally, throughout this chapter we assume that

1. the system is strictly overdetermined, i.e., $M > R$,
2. $\mathbf{H}$ and $\mathbf{S}^{\mathrm{T}}$ have full column rank,
3. the inputs have zero mean and are temporally coherent but mutually independent,
4. the additive noise is independent and identically distributed.

Temporal coherence means that the input signals are correlated with delayed versions of itself, which is typically the case for smooth signals and appears

frequently in practice. For example, both speech and biomedical signals such as ECG signals are temporally coherent. Using these assumptions and the structural information described above, the goal is to retrieve the original inputs $s_r(t)$ for $r \in \{1, \ldots, R\}$ from the observed mixtures. The assumption that $\mathbf{H}$ has full column rank allows that a few input-output paths have shorter FIR filters, but it is not allowed that a single source has a maximum filter delay that is lower than L to all outputs as this creates an all-zeros column in $\mathbf{H}$. Note that the assumption that $\mathbf{H}$ has full column rank implies that the system is strictly overdetermined.

Note that throughout most chapters of the thesis it is assumed that the number of inputs R and the maximum filter delay L are known. However, in practice this is not always the case and these values will often have to be estimated. One approach to do so has been presented in [138], which will be briefly repeated here. To retrieve R, one can increase the value of L' by one, which leads to an increase in rank of $\mathbf{X}$ of R. To retrieve the maximum filter delay as well, the rank of $\mathbf{X}$ (which equals RL_{tot}) can be combined with the value of R to retrieve L. Note that in noisy conditions, the exact rank cannot be used and the number of dominant singular values must be considered instead. If these parameters are not chosen correctly, the accuracy of the developed methods will drop. By using the data to estimate these parameters however, even wrongly estimated parameters may still results in a system model that approximates the behavior of the actual system, which in turn may still lead to useful output.

2.3 Deconvolution after tensorization

In [13, 12], a convolutive generalization of the popular SOBI-algorithm for instantaneous mixtures [10] is presented, as mentioned in the introduction. The SOBI algorithm works by stacking several lagged coveriance matrices of the instantaneous mixture $\mathbf{m}(t)$, such as $\mathbb{E}\left\{\mathbf{m}(t)\mathbf{m}^{\mathrm{H}}(t+\tau)\right\}$, in a tensor. For instantaneous mixtures, this resulting tensor admits a CPD of which the factors contain the mixing vectors of the mixture. More details on this method can be found in [10].

The convolutive generalization of the SOBI-method as presented in [13, 12] follows a similar approach. The methods starts by computing lagged second-order statistics of $\mathbf{x}(n)$ as defined above, for instance via the sample covariance matrix computation but with one lagged output, which can be stacked into a

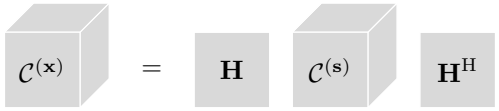


Figure 2.1: The unmixing step in convolutive ICA can be written as a block term decomposition in multilinear rank-$(L_{\mathrm{tot}}, L_{\mathrm{tot}}, \cdot)$ terms.

third-order tensor $\mathcal{C}^{(\mathbf{x})}$. The kth frontal slice of this tensor is then given by

$$\begin{aligned}\mathbf{C}_k^{(\mathbf{x})} &= \mathbb{E}\left\{\mathbf{x}(n)\mathbf{x}^{\mathrm{H}}(n+\tau_k)\right\} \in \mathbb{C}^{ML' \times ML'} \\ &= \mathbf{H}\,\mathbb{E}\left\{\mathbf{s}(n)\mathbf{s}^{\mathrm{H}}(n+\tau_k)\right\}\mathbf{H}^{\mathrm{H}} + \operatorname{diag}(\boldsymbol{\sigma}_v)\,\delta(\tau_k),\end{aligned}$$

in which $\operatorname{diag}(\boldsymbol{\sigma}_v)$ is the contribution of the additive noise term, which is only present for $\tau_k = 0$ under the assumption that the additive noise is i.i.d. Because the inputs are temporally coherent and mutually independent, it follows from (2.3) that the (lagged) covariance matrix $\mathbb{E}\left\{\mathbf{s}(t)\mathbf{s}^{\mathrm{H}}(t+\tau_k)\right\}$ will be block-diagonal. Consequently, the frontal slices of $\mathcal{C}^{(\mathbf{x})}$ can be jointly block-diagonalized [13]. Note that if the inputs were not temporally coherent, all lagged statistics would be zero and would contain no information. Alternatively, if the inputs were not mutually independent, the (lagged) covariance matrices of the inputs would be full matrices rather than block-diagonal.

From the joint block-diagonalization, $\mathbf{H}$ can be found up to two indeterminacies: (1) its block columns $\mathbf{H}_i$ can be arbitrarily permuted and (2) each block column $\mathbf{H}_i$ can be multiplied with a nonsingular matrix $\mathbf{E}_i \in \mathbb{C}^{L_{\mathrm{tot}} \times L_{\mathrm{tot}}}$. Though the authors did not realize it at the time, this simultaneous block-diagonalization is in fact a block-term decomposition in multilinear rank-$(L_{\mathrm{tot}}, L_{\mathrm{tot}}, \cdot)$ terms [31], as illustrated in Figure 2.1. Fitting this method in a broader framework for tensor-based convolutive ICA is a first (limited) contribution of this thesis. Algorithms for this block-term decomposition can be found in [1, 41, 113]. Because of the indeterminacies of joint block-diagonalization, computing the decomposition only allows us to retrieve the inputs up to a FIR filter [13]. One can deal with these filters using SIMO deconvolution techniques [90], which eventually return the inputs up to scaling and permutation.

In instantaneous ICA, all tensor-based methods compute statistics which are stored in a tensor. By subsequently decomposing this tensor, the mixing system and sources are extracted [10, 22]. Conceptually, the approach for convolutive systems above is exactly the same, which implies that this method nicely fits in a tensor-based framework for convolutive ICA.

2.4 Deconvolution before tensorization

One of the drawbacks of the method from the previous section lies in the computation time. By taking lagged outputs into account, more statistics have to be estimated. Moreover, joint block-diagonalization may take quite some time. By first deconvolving the outputs, we reduce the convolutive mixture to an instantaneous one, which can be solved more quickly. In this chapter, the main focus lies on the deconvolution step, which has to be both accurate and computationally inexpensive.

Subspace methods exploiting block-Toeplitz structure are well-established in signal and array processing [138, 139, 91, 124], and will be discussed in more depth in the next chapter. We briefly recall a technique from [124]. Compute the dominant row space of $\mathbf{X}$, which will be equal to the row space of $\mathbf{S}$ since $\mathbf{H}$ is assumed to have full column rank. Store the RL_{tot} basis vectors for the dominant subspace of $\mathbf{X}$ in the matrix $\mathbf{U} \in \mathbb{C}^{RL_{\text{tot}} \times N}$. We now have

$$\mathbf{U}^{\mathrm{T}} = \widetilde{\mathbf{S}}^{\mathrm{T}}\mathbf{M} = \left[\left(\widetilde{\mathbf{S}}^{(1)}\right)^{\mathrm{T}}, \ldots, \left(\widetilde{\mathbf{S}}^{(L_{\text{tot}})}\right)^{\mathrm{T}}\right]\mathbf{M}, \tag{2.4}$$

in which $\mathbf{M} \in C^{RL_{\text{tot}} \times RL_{\text{tot}}}$ and additive noise has been neglected for simplicity. Since $\widetilde{\mathbf{S}}$ has block-Toeplitz structure, equation (2.4) represents a block-Toeplitz factorization. Methods to solve this type of factorization can be found in [90, 92, 124]. We will use the result from [124], in which Lemma II.4 states that the decomposition in (2.4) is essentially unique if $\mathbf{M}$ and $\mathbf{T}$ have full column rank, in which $\mathbf{T} \in \mathbb{C}^{(N-1)\times R(L_{\text{tot}}+1)}$ is defined by

$$\mathbf{T} = \left[\overline{\left(\widetilde{\mathbf{S}}^{(1)}\right)^{\mathrm{T}}}, \ldots, \overline{\left(\widetilde{\mathbf{S}}^{(L_{\text{tot}})}\right)^{\mathrm{T}}}, \underline{\left(\widetilde{\mathbf{S}}^{(L_{\text{tot}})}\right)^{\mathrm{T}}}\right].$$

In this problem, essential uniqueness means that the mixture is deconvolved up to an instantaneous mixture. Mathematically, this implies that if the conditions hold, we obtain an input matrix estimate $\mathbf{K} \in \mathbb{C}^{RL_{\text{tot}} \times N}$ that has block-Toeplitz structure and that is related to the real inputs as follows:

$$\mathbf{K} = (\mathbf{I}_{L_{\text{tot}} \times L_{\text{tot}}} \otimes \mathbf{A})\,\widetilde{\mathbf{S}}, \tag{2.5}$$

in which $\mathbf{A} \in \mathbb{C}^{R\times R}$ is a nonsingular matrix.

To obtain the matrix $\mathbf{K}$ above, following procedure can be used. Starting from equation (2.4), the matrix $\mathbf{M}^{\mathrm{T}}$ can be partitioned in submatrices $\mathbf{M}_i^{\mathrm{T}} \in \mathbb{C}^{RL_{\text{tot}} \times R}$ such that

$$\mathbf{U}^{\mathrm{T}}\mathbf{M}_i^{\mathrm{T}} = \left(\widetilde{\mathbf{S}}^{(i)}\right)^{\mathrm{T}} \qquad \text{for} \quad i \in \{1, \ldots, L_{\text{tot}}\}.$$

By creating the stacked matrices

$$\mathbf{M}_{\text{tall}}^{T} = \left[\mathbf{M}_{1}^{\mathrm{T}}; \ldots; \mathbf{M}_{L_{\text{tot}}}^{\mathrm{T}}\right],$$

$$\widetilde{\mathbf{S}}_{\text{tall}}^{T} = \left[\left(\widetilde{\mathbf{S}}^{(1)}\right)^{\mathrm{T}}; \ldots; \left(\widetilde{\mathbf{S}}^{(L_{\text{tot}})}\right)^{\mathrm{T}}\right],$$

these equations can be jointly written as

$$\left(\mathbf{I}_{L_{\text{tot}}} \otimes \mathbf{U}^{\mathrm{T}}\right) \mathbf{M}_{\text{tall}}^{(T)} = \widetilde{\mathbf{S}}_{\text{tall}}^{(T)}.$$

Because of the Toeplitz structure of $\widetilde{\mathbf{S}}$, the stacked matrix $\widetilde{\mathbf{S}}_{\text{tall}}^{(T)}$ can be written as a product of Toeplitz basis matrices (see [124] for details) jointly represented by $\mathbf{E}$ in this text, and the generating signals $\mathbf{K}_{\text{gen}}$. Solving the resulting linear set of equations

$$\left(\mathbf{I}_{L_{\text{tot}}} \otimes \mathbf{U}^{\mathrm{T}}\right) \mathbf{M}_{\text{tall}}^{(T)} = \mathbf{E}\mathbf{K}_{\text{gen}},$$

then immediately yields the generating signals $\mathbf{K}_{\text{gen}}$ of the block-Toeplitz matrix $\widetilde{\mathbf{S}}$ up to multiplication with a nonsingular matrix $\mathbf{A} \in \mathbb{C}^{R \times R}$ if the conditions mentioned above hold.

Since this deconvolution step is deterministic, only few data samples are needed in principle. Reducing the number of used data samples leads to a faster deconvolution. However, we do need the full deconvolved signals to properly estimate the statistics for the resulting instantaneous ICA problem. This problem is solved by using only a part of the samples N_{part} in the deconvolution step. The method described above applied to the partial subspace $\mathbf{U}_{\text{part}} \in \mathbb{C}^{RL_{\text{tot}} \times N_{\text{part}}}$ leads to an input matrix estimate $\mathbf{K}_{\text{part}} \in \mathbb{C}^{RL_{\text{tot}} \times N_{\text{part}}}$. Since both matrices share the same row space, they are related through

$$\mathbf{U}_{\text{part}}{}^{\mathrm{T}} \mathbf{G} = \mathbf{K}_{\text{part}}{}^{\mathrm{T}}, \tag{2.6}$$

with $\mathbf{G} \in \mathbb{C}^{RL_{\text{tot}} \times RL_{\text{tot}}}$. The transformation matrix $\mathbf{G}$ can be computed from this equation and subsequently applied to the full subspace $\mathbf{U}$ to obtain the full-length deconvolved signals $\mathbf{K} = \mathbf{G}^{\mathrm{T}}\mathbf{U}$.

Minimum number of samples To see how many samples must be retained in theory, we derive a lower bound on the number of observations N_{part} such that (2.4) is essentially unique. For generic inputs, $\mathbf{T}$ will have full column rank if

$$N_{\text{part}} \geq R(L_{\text{tot}} + 1) + 1. \tag{2.7}$$

It then follows that the input matrix $\widetilde{\mathbf{S}}$ has full row rank. Since $\mathbf{H}$ is assumed to have full column rank, it follows that $\mathbf{M}$ also has full column rank and subsequently that the decomposition in (2.4) is essentially unique. Also note that the linear system (2.6) is overdetermined if this condition holds.

Determining the number of samples in practice The number of samples N_{part} is a trade-off between computational speed and accuracy of the deconvolution. Estimating $R^2L_{\text{tot}}^2$ parameters from the $N_{\text{part}}RL_{\text{tot}}$ equations in (2.6) can be done in least-squares sense. The standard deviation of the least-squares estimator is proportional to

$$\sigma_{\text{LS}} \propto \frac{1}{\sqrt{N_{\text{part}}RL_{\text{tot}} - R^2L_{\text{tot}}^2}}.$$

To get an idea of the (relative) accuracy, one can use this equation to assess the influence of reducing the number of observations N_{part}. For instance, if $R = 2$ and $L_{\text{tot}} = 4$, then reducing the number of observations from 1000 to 100 increases the standard deviation on the estimator with a factor 3.28, which may still be reasonable depending on the application.

Resolving the remaining mixture Once $\mathbf{K}$ is estimated, one can extract the first block-row $\mathbf{K}^{(1)}$, for which it follows from (2.5) that

$$\mathbf{K}^{(1)} = \mathbf{A}\widetilde{\mathbf{S}}^{(1)} \in \mathbb{C}^{R\times N}.$$

Since $\widetilde{\mathbf{S}}^{(1)}$ contains all sources without lags, $\mathbf{K}^{(1)}$ is an instantaneous mixture of independent sources. Any existing technique for instantaneous ICA can be used to separate this remaining mixture up to scaling and permutation of the original inputs [28, 71].

2.5 Numerical experiments

2.5.1 Synthetic data

In both experiments below, the synthetic inputs are generated by passing a complex signal with components randomly sampled from a uniform distribution over $[0, 1]$ through a convolutive filter, of which the 20 coefficients have been randomly sampled from a standard normal distribution. This procedure yields mutually independent but temporally coherent inputs. The actual system coefficients making up the convolutive mixture are randomly sampled from a standard normal distribution. All additive noise is Gaussian.

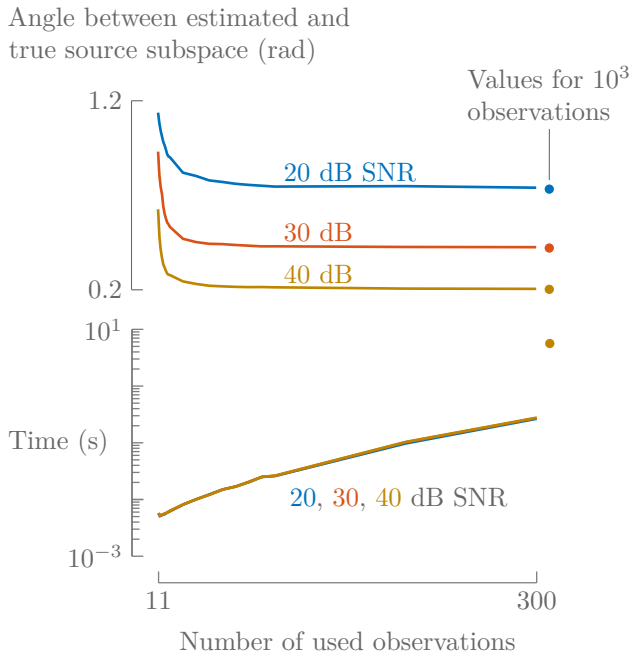


Figure 2.2: Using fewer observations enables faster deconvolution without sacrificing much accuracy for reasonable sample sizes. Here, the theoretical minimum number of observations needed is 11.

Effect of partial information for deconvolution

To illustrate the effect of partial information, consider a convolutive mixture with 4 outputs, 2 inputs and a maximum filter delay of 2. The available output signals have 10^3 samples. Following the approach outlined in Section 2.4, we estimate the input subspace using only part of the available data. The mean result over 100 experiments is shown in Figure 2.2. The theoretical minimum number of samples can be computed using equation (2.7) and is equal to 11 in this example. In the neighborhood of this theoretical minimum, the accuracy is low due to the big influence of noise. For a higher number of observations, the figure shows that using partial information enables a much faster deconvolution, with only little loss in accuracy when compared to the values for 10^3 observations. Note that the subspace estimates do not keep improving when more observations are taken into account. This can be explained by noting that we are trying to estimate the long space of a matrix, which cannot be estimated consistently in

noisy conditions [43]. However, in the absence of noise the estimate would be exact as the deconvolution step is deterministic.

Comparison of both strategies

To see whether it is better to deconvolve first and tensorize afterwards or the other way around, we compare both approaches. Consider a system with 4 outputs, 2 inputs, a maximum filter delay of $L = 1$ and signals of length 10^4. For the fast deconvolution presented in Section 2.4, 200 observations were used. After deconvolution, SOBI is used with the same time lags as its convolutive extension in this experiment (which are the first 20 nonzero lags). Figure 2.3 shows the mean performance over 1000 experiments of both approaches for various signal-to-noise ratio (SNR) values and for a varying number of available observations. The performance is expressed as the relative error on the inputs, after optimal permutation and scaling of the estimates.

The left part of the figure shows that the approach from [13], which starts by tensorizing, performs slightly better. However, this comes at the costs of a slower method. On average, the method takes 0.121 s on a standard laptop[1], whereas deconvolution before tensorization takes 0.061 s, both for signals of length 10^4. The difference in accuracy can be explained as follows: since the available signals are long, the lagged statistics can be estimated accurately. Therefore, tensorizing the data first has a denoising effect without much loss of accuracy. If this is the case, we expect the accuracy to drop when fewer data samples are available because the estimation error of the statistics will be larger. This is indeed the case, as illustrated in the right part of Figure 2.3, which has been simulated with 20 dB SNR. The approach starting with deconvolution is less influenced by the number of data samples because the deconvolution step is deterministic. The crossover point lies around 100 samples. So when few data points are available, and statistics cannot be estimated accurately, one should deconvolve first.

2.5.2 Application to rotating machines

Vibration analysis is often used in rotating machines to find faults such as bearing defects without stopping production [126]. From the frequencies of the machine vibrations, one can pinpoint which component is faulty. However, if several machines are working side by side, their vibrations may travel through the foundation and disturb measurements. This is where blind source separation

[1] Using Matlab on a laptop containing an Intel Core i7-4810MQ CPU.

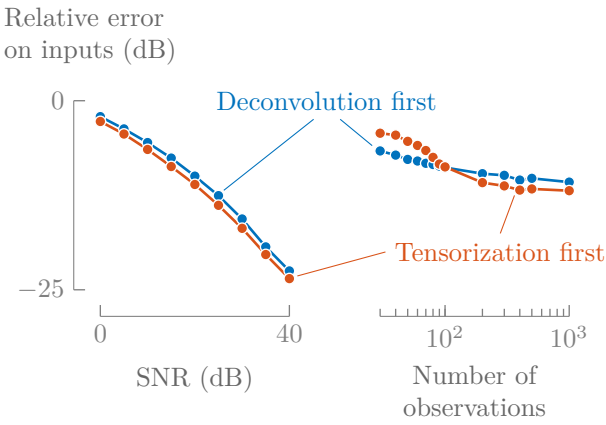


Figure 2.3: Using deconvolution or tensorization as the first step in the algorithm yields comparable results. The small difference between both approaches arises from the number of available observations.

comes in [60]. If the machines are sufficiently different, their vibrations will be statistically independent. This motivates the use of convolutive ICA techniques. To illustrate our method in this context, we simulate rotating machine vibrations using a model for gearbox vibrations used in [60][2]. The simulated vibrational signals are mixed convolutively with filters of maximum length $L = 3$ to obtain 5 outputs, to which Gaussian noise is added such that the SNR is 30 dB. From these outputs, the inputs are estimated using the method that first deconvolves and then tensorizes, using 200 observations for the deconvolution and SOBI with 20 nonzero time lags as instantaneous ICA method. The results were computed in 0.16 s on a standard laptop. The obtained relative error on the original vibration signals is −18.8 dB. Figure 2.4 shows that at 72 Hz, a small error is noticeable in the estimated input where the effect of the other input slightly bleeds through. Apart from this, the frequency peaks of both signals are cleanly separated.

[2]For completeness, we give all parameter values. First source: $a = 3$, $\mathbf{A} = [-0.2792, -1.7251, -0.2326]$, $f_m = 55$, $\boldsymbol{\phi}_i = [0.0027, 0.1546, 0.0808]$, $b = 2$, $\mathbf{B} = [1.8883, -0.8930]$, $f_p = 77$, $\boldsymbol{\phi}_j = [-0.3361, 0.288]$. Second source: $a = 3$, $\mathbf{A} = [-0.1479, -2.7398, -1.6482]$, $f_m = 45$, $\boldsymbol{\phi}_i = [0.2198, 0.1782, -0.0626]$, $b = 2$, $\mathbf{B} = [-1.6716, -1.0965]$, $f_p = 81$, $\boldsymbol{\phi}_j = [0.0842, -0.2716]$.

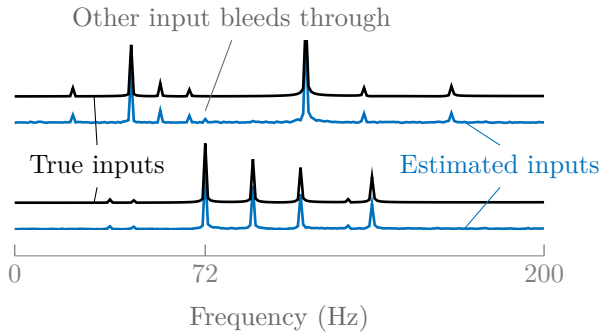


Figure 2.4: Apart from a slight influence at 72 Hz, the fast deconvolution approach followed by SOBI cleanly separates a convolutive mixture of vibrational signals stemming from rotational machines.

2.6 Conclusion

First, we linked an existing method for convolutive ICA with tensor-based methods. Next we contrasted this method with its opposite approach that starts by deconvolving the data before tensorizing. To do the deconvolution efficiently, a new subspace-based method was presented which uses only part of the available information. Through numerical experiments, this fast deconvolution was shown to be almost as accurate as methods using all information, while being much faster. Its effectiveness was illustrated in a simulation of rotational machine vibrations. Finally, we compared the speed and accuracy of the approach starting with tensorization and its opposite approach.

Chapter 3

Tensor decompositions with several block-Hankel factors and application in blind system identification

Abstract — Several applications in biomedical data processing, telecommunications or chemometrics can be tackled by computing a structured tensor decomposition. In this paper, we focus on tensor decompositions with two or more block-Hankel factors, which arise in blind multiple-input-multiple-output (MIMO) convolutive system identification. By assuming statistically independent inputs, the blind system identification problem can be reformulated as a Hankel structured tensor decomposition. By capitalizing on the available block-Hankel and tensorial structure, a relaxed uniqueness condition for this structured decomposition is obtained. This condition is easy to check, yet very powerful. The uniqueness condition also forms the basis for two subspace-based algorithms, able to blindly identify linear underdetermined MIMO systems with finite impulse response.

Reference — This chapter is a slightly adapted version of [147]. Starting from [121], the candidate did the research and wrote the article under the guidance of the coauthors.

3.1 Introduction

Tensor decompositions are widespread in signal processing [81, 25]. One of the most popular decompositions is the canonical polyadic decomposition (CPD), also called PARAFAC or CANDECOMP, which writes a tensor as a linear combination of a minimal number of rank-1 terms. Computing a CPD is often done using optimization-based methods, such as alternating least squares or all-at-once optimization methods, see e.g. [113, 4, 104, 130]. Because the optimization problems arising in tensor decompositions are generally nonconvex, these methods may converge to local minima depending on the initialization. By contrast, algebraic methods are guaranteed to find the exact solution in the noiseless case, for instance using the generalized eigenvalue decomposition [30, 87, 51]. Moreover, algebraic methods are usually faster than optimization-based methods for reasonably sized tensors. The main downside of algebraic methods is that they only return the exact solution in the noiseless case. For high noise levels, optimization-based algorithms are more accurate because they directly fit a CPD model to the noisy data. Still, algebraic methods can be useful in this case to provide a good initial value at low cost. The reasons above motivate why algebraic methods for various (structured) tensor decompositions have been developed, e.g. in [63, 30, 116]. In particular, an algebraic method for tensors with a block-Toeplitz factor has been introduced in [124, 119]. Here, we will extend this result and consider algebraic methods for tensors with at least two block-Hankel factors. Such tensors arise in blind system identification [59]. More recently, tensors with block-Hankel structured factors have also appeared in machine learning [69], and structured CPDs have been studied in [62].

As a first contribution, we will formulate a uniqueness condition for tensor decompositions with block-Hankel factor matrices. By exploiting the available structure, a more relaxed yet easy-to-check uniqueness condition is obtained. Our second contribution comprises the development of two algorithms for computing tensor decompositions with several block-Hankel factor matrices.

Since blind system identification (BSI) is an important application of tensor decompositions with block-Hankel structured factor matrices, we will use it as motivating example throughout this paper. BSI tries to identify a system using only output values. To make this problem feasible, assumptions have to be imposed on the inputs or on the system itself [132]. Depending on the application, some possible assumptions are statistically independent components [28], finite alphabet [139], or constant modulus [133, 137]. BSI with statistically independent inputs is closely related to independent component analysis (ICA), a well-known technique for extracting independent components from a set of mixtures. The slight conceptual difference between BSI and ICA is that the former focuses on estimating the mixing system whereas the latter tries to

reconstruct the inputs. For overdetermined systems this amounts to the same, but for underdetermined systems, knowledge of the mixing system does not suffice to reconstruct the inputs.

ICA and BSI for instantaneous mixtures have applications in various domains, such as biomedicine [28, 36], image processing [70], and communication technology [28]. Tensor-based methods for instantaneous mixtures are well established, see [28] and references therein. In wireless telecommunications, audio processing and several other domains, the convolutive mixture model is often more appropriate [2, 28]. For this model, tensor-based algorithms have received relatively little attention [59, 131, 164]. Only some of the existing approaches formulate uniqueness conditions based on Kruskal's result [131, 164, 84]. We will show that by exploiting the available structure, our uniqueness conditions are more relaxed than the previously obtained conditions.

The subspace-based methods presented in this paper are also applicable to BSI. We mention that subspace-based methods for convolutive ICA have already been proposed [165, 88]. However, the methods in [165, 88] essentially ignore the tensor structure of the problem. For this reason, these methods can only handle the overdetermined case ("more outputs than inputs") while the methods here can also deal with the underdetermined case ("fewer outputs than inputs") and are guaranteed to find the decomposition in the exact case.

The remainder of the introduction will present the used notation followed by a brief discussion of the CPD. To motivate the study of tensor decompositions with block-Hankel factors, we will review an existing cumulant-based approach for blind identification of multiple-input-multiple-output (MIMO) finite impulse response (FIR) channels [59] in Section 3.2. Section 3.3 then provides new uniqueness results for the resulting tensor decomposition and Section 3.4 presents new subspace methods for the actual MIMO FIR identification. In Section 3.5 some numerical experiments are conducted.

3.1.1 Notation

Scalars, vectors, matrices and tensors are denoted by lower case (a), lower case boldface ($\mathbf{a}$), upper case boldface ($\mathbf{A}$) and upper case calligraphic letters ($\mathcal{A}$), respectively. The rth column vector of $\mathbf{A}$ is denoted by $\mathbf{a}_r$. The symbols $\otimes$ and $\odot$ denote the Kronecker and Khatri–Rao product, respectively. The outer product of N vectors $\mathbf{a}^{(n)} \in \mathbb{C}^{I_n}$ is denoted by $\mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$, such that $(\mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)})_{i_1,i_2,\ldots,i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}$. Column-wise vectorization is denoted by $\mathrm{Vec}\,(\cdot)$.

The transpose, conjugate, conjugate-transpose and column space of a matrix are denoted by $(\cdot)^{\mathrm{T}}$, $(\cdot)^*$, $(\cdot)^H$ and $\operatorname{range}(\cdot)$, respectively. The symbols $(\cdot)^{\perp}$ and $\dim(\cdot)$ denote the orthogonal complement and dimension of a subspace.

The identity matrix and all-ones vector are denoted by $\mathbf{I}_R \in \mathbb{C}^{R\times R}$ and $\mathbf{1}_R \in \mathbb{C}^R$, respectively. The all-zeros vector and matrix are denoted by $\mathbf{0}_R \in \mathbb{C}^R$ and $\mathbf{0}_{R\times S} \in \mathbb{C}^{R\times S}$.

Finally, let $\mathbf{K} = \left[\left(\mathbf{K}^{(1)}\right)^{\mathrm{T}}, \ldots, \left(\mathbf{K}^{(N)}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{MN\times P}$, where $\mathbf{K}^{(n)} \in \mathbb{C}^{M\times P}$, then

$$\overline{\mathbf{K}} = \left[\left(\mathbf{K}^{(2)}\right)^{\mathrm{T}}, \ldots, \left(\mathbf{K}^{(N)}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{M(N-1)\times P},$$

$$\underline{\mathbf{K}} = \left[\left(\mathbf{K}^{(1)}\right)^{\mathrm{T}}, \ldots, \left(\mathbf{K}^{(N-1)}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{M(N-1)\times P}.$$

In other words, $\overline{\mathbf{K}}$ and $\underline{\mathbf{K}}$ are obtained by deleting the top and bottom block row of $\mathbf{K}$, respectively.

3.1.2 Canonical Polyadic Decomposition

A rank-1 tensor $\mathcal{X} \in \mathbb{C}^{I_1\times\cdots\times I_N}$ is defined as a tensor which can be written as the outer product of non-zero vectors $\mathbf{u}^{(n)} \in \mathbb{C}^{I_n}$. Decompositions into a sum of rank-1 terms are called Polyadic Decompositions (PDs):

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(N)}. \tag{3.1}$$

The rank of a tensor $\mathcal{X}$ is equal to the minimal number of rank-1 tensors that yield $\mathcal{X}$ in a linear combination. Assume that the rank of $\mathcal{X}$ is R, then (3.1) is called the Canonical PD of $\mathcal{X}$. The vectors $\{\mathbf{u}_r^{(n)}\}$ are often stacked into the matrices

$$\mathbf{U}^{(n)} = \begin{bmatrix}\mathbf{u}_1^{(n)} & \cdots & \mathbf{u}_R^{(n)}\end{bmatrix} \in \mathbb{C}^{I_n\times R},$$

which are called the factor matrices of the PD of $\mathcal{X}$. Expression (3.1) can then be compactly written as

$$\mathcal{X} = \left[\!\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}\right]\!\!\right].$$

The rank-1 tensors in (3.1) can be arbitrarily permuted without changing the decomposition. The different factor vectors of a rank-1 tensor can also be arbitrarily scaled provided that the overall rank-1 term remains the same. We say that the CPD is unique when it is only subject to these indeterminacies.

Regarding uniqueness of the CPD in this paper, it suffices to consider results for third-order tensors of which the third factor matrix $\mathbf{U}^{(3)}$ has full column rank. For this problem the following easy-to-check uniqueness result will be used.

Theorem 3.1.1. *Consider the PD of* $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ *in* (3.1). *If*

$$\begin{cases} \mathbf{U}^{(3)} \text{ has full column rank } R, \\ \mathcal{C}_2\left(\mathbf{U}^{(1)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(2)}\right) \text{ has full column rank,} \end{cases}$$

then the rank of $\mathcal{X}$ *is* R *and the CPD of* $\mathcal{X}$ *is unique [74, 30, 49, 50].*

In this theorem, $\mathcal{C}_k(\mathbf{A})$ is defined as the kth compound matrix of an $I \times R$ matrix $\mathbf{A}$ [49], which is the $\binom{I}{k} \times \binom{R}{k}$ matrix containing the determinants of all $k \times k$ submatrices of $\mathbf{A}$. The determinants are arranged such that the submatrix index sets are in lexicographic order.

3.2 Blind System Identification

Consider a linear time invariant FIR system with R inputs and M outputs, having $L+1$ coefficients per input-output channel. Asssume that the mth output at the nth sample period can be written as

$$y_m(n) = \sum_{r=1}^{R} \sum_{l=0}^{L} h_{mr}(l) s_r(n-l) + v(n),$$

in which $\{s_r(n)\}$ are unknown input signals, $\{h_{mr}(n)\}$ are unknown channel coefficients and $v(n)$ is an additive noise term.

The goal of BSI is to determine $\{h_{mr}(n)\}$ based on the observed data sequence $\{y_m(n)\}$. Here, we make the following assumptions:

1) The signals $\{s_r(n)\}$ are zero mean, non-Gaussian, independent and identically distributed (i.i.d.) stationary processes. The input signals are mutually statistically independent.

2) The additive noise $v(n)$ has zero mean, is Gaussian and is independent of the input signals.

As in [88, 59, 58] we work with the fourth-order spatio-temporal cumulants of the output signals:

$$\begin{aligned} & c_{m_1 m_2 m_3 m_4}(l_1, l_2, l_3) \\ & := \mathrm{Cum}\left[y^*_{m_1}(n), y_{m_2}(n+l_1), y^*_{m_3}(n+l_2), y_{m_4}(n+l_3)\right] \\ & = \sum_{r=1}^{R} \gamma_r \sum_{l=0}^{L} h_{m_1 r}(l)^* h_{m_2 r}(l+l_1) h_{m_3 r}(l+l_2)^* h_{m_4 r}(l+l_3) \end{aligned}$$

in which $\gamma_r = \mathrm{Cum}\left[s^*_r(n), s_r(n), s^*_r(n), s_r(n)\right]$, the subscripts $m_1, m_2, m_3, m_4 \in \{1, \ldots, M\}$ and $l_1, l_2, l_3 \in \{-L, \ldots, L\}$, and $h_{m_n r}(l) = 0$, $\forall l \notin \{0, \ldots, L\}$. Because the fourth-order cumulant of a Gaussian signal is zero, the additive noise does not affect this tensor in theory. We do mention that in a real-life setting, Gaussian noise will still have an influence because the signals have finite length. Computing this cumulant leads to a 7th-order tensor $\mathcal{C}$ with size M in the first four modes and size $2L+1$ in the last three. Now permute this tensor by ordering the 7 dimensions as $\{1, 2, 5, 3, 6, 4, 7\}$. Next, reshape this result into a fourth-order tensor by combining the modes $\{2, 5\}$, $\{3, 6\}$ and $\{4, 7\}$ to reveal the structure [59]. Doing so yields the fourth-order tensor $\mathcal{T} \in \mathbb{C}^{M \times M(2L+1) \times M(2L+1) \times M(2L+1)}$, which can be written as

$$\mathcal{T} = \sum_{r=1}^{R} \gamma_r \sum_{l=0}^{L} \mathbf{p}_r^{(l)*} \circ \mathbf{h}_r^{(l)} \circ \mathbf{h}_r^{(l)*} \circ \mathbf{h}_r^{(l)}, \tag{3.2}$$

in which $\mathbf{p}_r^{(l)}$ is the rth column of $\mathbf{P}^{(l)}$, defined as

$$\mathbf{P}^{(l)} = \begin{bmatrix} h_{11}(l) & \cdots & h_{1R}(l) \\ \vdots & \ddots & \vdots \\ h_{M1}(l) & \cdots & h_{MR}(l) \end{bmatrix} \in \mathbb{C}^{M \times R},$$

for $l \in \{0, \ldots, L\}$. The vector $\mathbf{h}_r^{(l)}$ denotes the rth column of $\mathbf{H}^{(l)}$, which is given by

$$\mathbf{H}^{(l)} = \begin{bmatrix} \mathbf{0}_{M(L-l),R} \\ \mathbf{P} \\ \mathbf{0}_{Ml,R} \end{bmatrix} \in \mathbb{C}^{M(2L+1) \times R}, \tag{3.3}$$

with $l \in \{0, \ldots, L\}$. The matrix $\mathbf{P}$ is defined by

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{(0)} \\ \vdots \\ \mathbf{P}^{(L)} \end{bmatrix} \in \mathbb{C}^{M(L+1) \times R}, \tag{3.4}$$

and contains all system parameters. More precisely, the rth column of $\mathbf{P}$ contains the parameters related to the rth system input.

We use the simplified notation

$$\mathcal{T} = [\![\mathbf{G}^*, \mathbf{H}, \mathbf{H}^*, \mathbf{H}]\!], \tag{3.5}$$

in which

$$\mathbf{G} = \begin{bmatrix} \mathbf{P}^{(0)}\Gamma & \cdots & \mathbf{P}^{(L)}\Gamma \end{bmatrix} \in \mathbb{C}^{M \times R(L+1)}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}^{(0)} & \cdots & \mathbf{H}^{(L)} \end{bmatrix} \in \mathbb{C}^{M(2L+1) \times R(L+1)},$$

and the matrix $\Gamma = \operatorname{diag}(\gamma_1, \cdots, \gamma_r)$. Figure 3.1 illustrates the banded block-Hankel structure of $\mathbf{H}$.

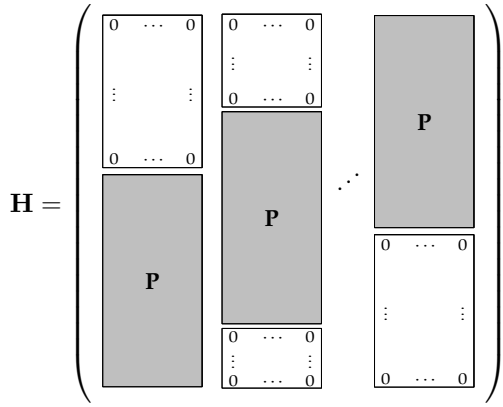


Figure 3.1: Structure of the factor matrix $\mathbf{H}$.

Let us now unfold $\mathcal{T}$ by first constructing the matrix slices $\mathbf{T}^{(i_1,i_2)} \in \mathbb{C}^{M(2L+1) \times M(2L+1)}$ so that $t^{(i_1,i_2)}_{i_3 i_4} = t_{i_1 i_2 i_3 i_4}$. Define $\mathbf{t}^{(i_1,i_2)} = \operatorname{Vec}\left(\left(\mathbf{T}^{(i_1,i_2)}\right)^{\mathrm{T}}\right)$, then decomposition (3.5) has the following matrix representation

$$\begin{aligned} \mathbf{T} &= \left[\mathbf{t}^{(1,1)}, \ldots, \mathbf{t}^{(1,M(2L+1))}, \mathbf{t}^{(2,1)}, \ldots, \mathbf{t}^{(M,M(2L+1))}\right] \\ &= (\mathbf{H}^* \odot \mathbf{H})\, \mathbf{C}^{\mathrm{T}}, \end{aligned} \tag{3.6}$$

where $\mathbf{C} = (\mathbf{G}^* \odot \mathbf{H}) \in \mathbb{C}^{M^2(2L+1) \times R(L+1)}$.

In the complex case, equation (3.6) is a CPD with partial Hermitian symmetry. As explained in [118], this structure can be exploited by constructing the

augmented matrix

$$\begin{aligned}\mathbf{Y} &:= \left[\mathbf{T}, \widetilde{\mathbf{t}}^{(1,1)}, \widetilde{\mathbf{t}}^{(1,2)}, \ldots, \widetilde{\mathbf{t}}^{(M,M(2L+1))}\right] \\ &= (\mathbf{H}^* \odot \mathbf{H})\, \mathbf{D}^{\mathrm{T}} \in \mathbb{C}^{M^2(2L+1)^2 \times 2M^2(2L+1)}, \end{aligned} \tag{3.7}$$

in which $\widetilde{\mathbf{t}}^{(i,j)} = \mathrm{Vec}\left(\left(\mathbf{T}^{(i,j)}\right)^*\right)$ and

$$\mathbf{D} = \begin{bmatrix} \mathbf{G}^* \odot \mathbf{H} \\ \mathbf{G} \odot \mathbf{H}^* \end{bmatrix} \in \mathbb{C}^{2M^2(2L+1) \times R(L+1)}. \tag{3.8}$$

By considering $\mathbf{Y}$ instead of $\mathbf{T}$, more relaxed uniqueness conditions may be obtained. Note that exploiting partial Hermitian structure is only possible in the complex case. When dealing with real data, $\mathbf{Y}$ is equal to matrix $\mathbf{T}$ from equation (3.6).

For simplicity, we present equations (3.6) and (3.7) as equalities. However, in practice, this model will only hold approximately due to estimation errors of the higher-order statistics. Moreover, even though the cumulant of a Gaussian signal is zero in theory, the additive noise will still have an influence since we are dealing with finite signal lengths. These effects cannot be avoided, but they are partially alleviated by solving decompositions and linear systems in a least-squares sense.

At this point, we stress the importance of the stated assumptions at the start of this section. If the source signals would for instance be Gaussian, their fourth-order cumulant would be zero and thus would not allow any decompositions. If the source signals were not i.i.d. but temporally coherent, the resulting tensor decomposition would be a block-term decomposition rather than a CPD. If the input signals were not mutually independent, the fourth-order statistics of the input signals would lead to a full tensor rather than a diagonal one. The stated conditions are thus carefully selected such that the blind system identification problem can be written as a structured CPD of a tensor. Finally, if the additive noise was not Gaussian, the resulting cumulant tensor would be perturbed by noise, which will either have to be explicitly extracted or not taken into account in order to avoid influencing the resulting estimates.

Based on the matrix representation $\mathbf{Y}$ we will now present a new uniqueness condition for the PD of $\mathcal{T}$ in (3.2) taking the low-rank Khatri–Rao structure and the block-Hankel structure of the problem into account. We will use $\mathcal{T}$ to derive all results, but the results can easily be generalized to any tensor that can be written as a CPD having at least two block-Hankel structured factors.

3.3 Identifiability condition for BSI

In the instantaneous mixture case, which is a simplification of our convolutive model, each output is a linear combination of the different inputs without time lags. If R inputs are mixed instantaneously, the fourth-order cumulant of the outputs is a tensor of rank R [28, 35, 23, 27, 18]. The contribution of each input is contained in a rank-1 term. By decomposing the cumulant tensor into R rank-1 terms, i.e. computing the CPD, the rank-1 tensors can be retrieved up to a permutation of the terms and scaling of the factor matrices, as explained in Section 3.1.2.

Let us now consider the convolutive mixture case. Due to the time shifts in the convolution, the spatio-temporal cumulant that has to be computed is larger. Moreover, the fundamental blocks associated with each of the inputs are no longer rank-1 tensors, but tensors with Hankel structured factor matrices, as can be seen in equation (3.2). Note that all terms in the second summation are contributed by the same input and that $\mathbf{h}_r^{(l)}$ is the rth column of the $(l+1)$th submatrix of $\mathbf{H}$ in Figure 3.1, for $l = 0, 1, \ldots, L$. Mathematically, the fundamental Hankel-structured tensors are given by

$$\mathcal{T}^{(r)} = \gamma_r \sum_{l=0}^{L} \mathbf{p}_r^{(l)*} \circ \mathbf{h}_r^{(l)} \circ \mathbf{h}_r^{(l)*} \circ \mathbf{h}_r^{(l)} ,$$

for $r = 1, 2, \ldots, R$. Each of the R fundamental tensors contributed by an input has rank $L+1$. The difference between the instantaneous case and the convolutive case is illustrated in Figure 3.2. We define the Hankel constrained rank of $\mathcal{T}$ as the minimal number of Hankel structured tensors that yield $\mathcal{T}$ in a linear combination. The Hankel-structured decomposition is again subject to indeterminacies, which are similar to the CPD case. The Hankel-structured tensors can be arbitrarily permuted and the factors within the Hankel-structured tensors can be arbitrarily scaled provided that the overall term remains the same. We say that the Hankel-structured decomposition of $\mathcal{T}$ in (3.2) is unique if all parameter matrices $\widehat{\mathbf{P}}$ satisfying (3.2) are related via

$$\widehat{\mathbf{P}} = \mathbf{P}\Pi\Delta,$$

where Π is a permutation matrix and Δ is a nonsingular diagonal scaling matrix. Note that the parameters γ_r depend on the scaling matrices Δ, since the overall Hankel-structured tensors have to remain the same. For simplicity of terminology, we will say that $\mathbf{P}$ is unique in this case. The following subsections will present a deterministic and generic uniqueness condition for this Hankel-structured tensor decomposition (3.2) up to the mentioned indeterminacies.

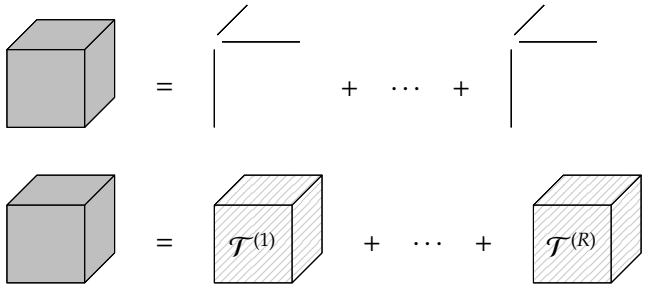


Figure 3.2: Conceptual difference between a decomposition into rank-1 terms (top) and a decomposition into Hankel structured tensor terms (bottom). The hatched tensors have Hankel-structured factors.

3.3.1 Deterministic uniqueness condition

Obtaining new uniqueness conditions will be a two-step process. First, Lemma 3.3.1 will capitalize on the block-Hankel structure to reduce the convolutive mixture to an instantaneous one. Second, Proposition 3.3.2 will exploit the CPD structure to find conditions for the actual identification of the system.

Because $\mathbf{H}$ has block-Hankel structure, it holds that

$$\overline{\mathbf{H}^{(i)}} = \underline{\mathbf{H}^{(i+1)}},$$

and consequently it follows that

$$\overline{\mathbf{H}^{(i)}} \odot \overline{\mathbf{H}^{(i)}} = \underline{\mathbf{H}^{(i+1)}} \odot \underline{\mathbf{H}^{(i+1)}}. \tag{3.9}$$

We can now apply an existing theorem to exploit the shift-invariance shown in (3.9), as found in e.g. [90, 92]:

Lemma 3.3.1. *Consider the cumulant tensor $\mathcal{T}$ in* (3.2) *and the augmented matrix representation $\mathbf{Y} = (\mathbf{H}^* \odot \mathbf{H}) \cdot \mathbf{D}^T$, defined in* (3.7)*. Define the matrix $\mathbf{N} = \left[\underline{\mathbf{H}^{(0)}} \quad \underline{\mathbf{H}^{(1)}} \quad \cdots \quad \underline{\mathbf{H}^{(L)}} \quad \overline{\mathbf{H}^{(L)}}\right]$. Assume that*

$$\begin{cases} \mathbf{D} \text{ in } (3.8) \text{ has full column rank,} \\ \mathbf{N}^* \odot \mathbf{N} \text{ has full column rank.} \end{cases}$$

Let $\mathbf{K} \in \mathbb{C}^{M^2(2L+1)^2 \times R(L+1)}$ be another matrix with the same block-Hankel structure as $\mathbf{H}^* \odot \mathbf{H}$, then range$(\mathbf{K})$ = range$(\mathbf{H}^* \odot \mathbf{H})$ = range$(\mathbf{Y})$ if and only if $(\mathbf{H}^* \odot \mathbf{H}) = \mathbf{K}(\mathbf{I}_{L+1} \otimes \mathbf{A})$, in which $\mathbf{A} \in \mathbb{C}^{R \times R}$ is a nonsingular matrix.

Note that the definition of the matrix $\mathbf{N}$ is motivated by the conditions for essential uniqueness of the block-Hankel decomposition of $\mathbf{Y}$. The mathematical justification for this definition can be found in [90], though formulated in a slightly different way.

Using the lemma above, we can now formulate a uniqueness condition for the Hankel-structured tensor decomposition in a MIMO FIR identification context.

Proposition 3.3.2. *Consider the PD of $\mathcal{T}$ in* (3.2) *with Hankel-constrained rank R. If*

$$\begin{cases} \mathbf{D} \text{ in } (3.8) \text{ has full column rank}, & (3.10a) \\ \mathbf{N}^* \odot \mathbf{N} \text{ has full column rank}, & (3.10b) \\ \mathcal{C}_2(\mathbf{P}^*) \odot \mathcal{C}_2(\mathbf{P}) \text{ has full column rank}, & (3.10c) \end{cases}$$

then the Hankel constrained rank of $\mathcal{T}$ is R and the Hankel-structured decomposition of $\mathcal{T}$ is unique.

Proof. Assume there exists an alternative solution with factor matrix $\widehat{\mathbf{H}} \in \mathbb{C}^{M(2L+1)\times R(L+1)}$ that has the same block-Hankel structure as $\mathbf{H}$. To prove uniqueness, it suffices to show that $\mathbf{P}$ is unique under the conditions above.

Since $\mathbf{N}^* \odot \mathbf{N}$ has full column rank, as assumed in (3.10b), it follows that $\mathbf{H}^* \odot \mathbf{H}$ also has full column rank. Let $\left(\widehat{\mathbf{H}}, \widehat{\mathbf{D}}\right)$, generating a tensor that has Hankel-constrained rank $\widehat{R}$, be an alternative solution such that

$$\mathbf{Y} = (\mathbf{H}^* \odot \mathbf{H})\,\mathbf{D}^{\mathrm{T}} = \left(\widehat{\mathbf{H}}^* \odot \widehat{\mathbf{H}}\right)\widehat{\mathbf{D}}^{\mathrm{T}}. \tag{3.11}$$

Since $\mathbf{H}^* \odot \mathbf{H}$ and $\mathbf{D}$ have full column rank, we know from (3.11) that $R = \widehat{R}$, that $\widehat{\mathbf{H}}^* \odot \widehat{\mathbf{H}}$ and $\widehat{\mathbf{D}}$ have full column rank and that $\operatorname{range}(\mathbf{H}^* \odot \mathbf{H}) = \operatorname{range}\left(\widehat{\mathbf{H}}^* \odot \widehat{\mathbf{H}}\right)$. The latter property together with Lemma 3.3.1 then implies that

$$\left(\widehat{\mathbf{H}}^* \odot \widehat{\mathbf{H}}\right) = (\mathbf{H}^* \odot \mathbf{H})\left(\mathbf{I}_{L+1} \otimes \mathbf{A}^{\mathrm{T}}\right), \tag{3.12}$$

in which $\mathbf{A} \in \mathbb{C}^{R\times R}$ is nonsingular. Now define

$$\mathbf{Z} = \left[\mathbf{Z}^{(0)}, \ldots, \mathbf{Z}^{(L)}\right] = (\mathbf{H}^* \odot \mathbf{H}),$$

$$\widehat{\mathbf{Z}} = \left(\widehat{\mathbf{H}}^* \odot \widehat{\mathbf{H}}\right).$$

The submatrices of $\mathbf{Z}$ take the following form:

$$\mathbf{Z}^{(0)} = \begin{bmatrix} \mathbf{0}_{ML,R} \\ \mathbf{P}^* \end{bmatrix} \odot \begin{bmatrix} \mathbf{0}_{ML,R} \\ \mathbf{P} \end{bmatrix} \in \mathbb{C}^{M^2(2L+1)^2 \times R},$$

$$\mathbf{Z}^{(l)} = \begin{bmatrix} \mathbf{0}_{M(L-l),R} \\ \mathbf{P}^* \\ \mathbf{0}_{Ml,R} \end{bmatrix} \odot \begin{bmatrix} \mathbf{0}_{M(L-l),R} \\ \mathbf{P} \\ \mathbf{0}_{Ml,R} \end{bmatrix}.$$

Using the relations above, the first block column of (3.12) can be written as

$$\widehat{\mathbf{Z}}^{(0)} = \mathbf{Z}^{(0)}\mathbf{A}^{\mathrm{T}} = \left(\begin{bmatrix} \mathbf{0}_{ML,R} \\ \mathbf{P}^* \end{bmatrix} \odot \begin{bmatrix} \mathbf{0}_{ML,R} \\ \mathbf{P} \end{bmatrix} \right) \mathbf{A}^{\mathrm{T}}.$$

Because the expression above is a matricized CPD [81], the matrix $\widehat{\mathbf{Z}}^{(0)}$ can be seen as an unfolded tensor

$$\widehat{\mathcal{Z}}^{(0)} = \left[\!\!\left[\begin{bmatrix} \mathbf{0}_{ML,R} \\ \mathbf{P}^* \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{ML,R} \\ \mathbf{P} \end{bmatrix}, \mathbf{A} \right]\!\!\right].$$

Because we are not interested in the zero parts of this expression, we extract the nonzero parts, which leads to a reduced tensor $\widehat{\mathcal{Z}}^{(0)}_{\mathrm{red}} = \sum_{r=1}^{R} \mathbf{p}_r^* \circ \mathbf{p}_r \circ \mathbf{a}_r$. From assumption (3.10c) and Theorem 3.1.1 it follows that the CPD of $\widehat{\mathcal{Z}}^{(0)}_{\mathrm{red}}$ is unique. This then means that $\mathbf{P}$ is unique.

We can thus conclude that if assumptions (3.10a)–(3.10c) are satisfied, the Hankel constrained rank is R and the Hankel-structured decomposition is unique. □

Note that in the proof above, the banded property of $\mathbf{H}$ has not been used, nor has the symmetry of $\mathcal{T}$. This implies that the uniqueness analysis so far holds for any tensor admitting a CPD with two block-Hankel structured factors. In general, the generating blocks $\mathbf{P}$ may differ in different modes and the entries of $\mathbf{Z}^{(0)}$ above and below $\mathbf{P}$ may be nonzero. In general, we allow $\widehat{\mathbf{Z}}^{(0)}$ of the form

$$\widehat{\mathbf{Z}}^{(0)} = \mathbf{Z}^{(0)}\mathbf{A}^{\mathrm{T}} = \left(\begin{bmatrix} \mathbf{K}^* \\ \mathbf{P}_a^* \end{bmatrix} \odot \begin{bmatrix} \mathbf{L} \\ \mathbf{P}_b \end{bmatrix} \right) \mathbf{A}^{\mathrm{T}}.$$

Such a matrix $\widehat{\mathbf{Z}}^{(0)}$ can again be seen as an unfolded tensor $\widehat{\mathcal{Z}}^{(0)}_{\mathrm{red}} = \sum_{r=1}^{R} (\mathbf{p}_a)_r^* \circ (\mathbf{p}_b)_r \circ \mathbf{a}_r$, extended with some extra rows. The conditions in Proposition 3.3.2 can be easily generalized to take into account that the generating submatrices of the block-Hankel factors in different modes may differ. If these generalized conditions hold, it follows that $\mathbf{P}_a$ and $\mathbf{P}_b$ are unique.

3.3.2 Generic uniqueness results

We study the generic properties of the uniqueness conditions stated in Proposition 3.3.2. We say that the Hankel constrained CPD of $\mathcal{T}$ in (3.2) is generically unique if the set of parameters $\mathbf{P}$ and $\boldsymbol{\gamma}$ that generate a tensor $\mathcal{T}$ for which the Hankel-structured decomposition is not unique is of Lebesgue measure zero.

For each of the rank conditions (3.10a)–(3.10c) in Proposition 3.3.2 we have to check when they are generically satisfied. For a square matrix, checking whether it has full rank comes down to checking whether its determinant is nonzero. Since the determinant is a polynomial expression in its parameters, we are effectively checking if an analytic function is nonzero. For any tall matrix $\mathbf{A}$, the same reasoning can be applied to all possible square submatrices of maximal size, since the rank of a matrix equals the size of the largest non-vanishing minor. Consequently, to check whether a tall matrix has full column rank one can simply check if any of the square submatrices of maximal size have full rank. As explained above, this is again equivalent to checking whether an analytic function is nonzero. To check the full-rank condition in the generic case, we resort to the following tool (e.g. [65]).

Lemma 3.3.3. *Given an analytic function $f : \mathbb{C}^n \to \mathbb{C}$. If there exists an element $\mathbf{x} \in \mathbb{C}^n$ such that $f(\mathbf{x}) \neq 0$, then the set $\{\, \mathbf{x} \mid f(\mathbf{x}) = 0 \,\}$ is of Lebesgue measure zero.*

This lemma essentially says that if we find one random example of a matrix having full rank, we may conclude that almost all matrices with the same structure will have full rank. Consequently, it suffices to numerically check the rank conditions for just one random example. We constructed random matrices $\mathbf{D}$, $\mathbf{N}$ and $\mathbf{P}$ that have the structure specified in Proposition 3.3.2 and then numerically checked for which values of R the rank conditions in the proposition hold. All results are shown in Table 3.1.

Combining all conditions, the upper bound for which Proposition 3.3.2 generically guarantees uniqueness of the Hankel-structured decomposition of $\mathcal{T}$ is given in Table 3.2. Note that the values in Table 3.2 coincide with those of condition (3.10a). To put it differently, the full column rank requirement of $\mathbf{D}$ is the most restrictive one. To compare the obtained uniqueness bound with existing results, the table also shows results from [131, 164]. Roughly speaking, in [131] it was observed using a slightly different tensor that, if

$$M(L+1) \geq R, \tag{3.13}$$

then the MIMO FIR parameter matrix $\mathbf{P}$ in (3.4) is generically unique. A frequency domain approach for convolutive ICA based on fourth-order cumulants

Table 3.1: For a given pair (M, L) the entry corresponds to the maximal value of R for which conditions (3.10a)–(3.10c) hold.

			L			
			2	3	4	5
Condition (3.10a)	M	2	12	13	13	14
		3	27	29	30	31
		4	48	52	54	56
		5	75	81	85	87
Condition (3.10b)	M	2	14	24	34	45
		3	31	54	78	102
		4	56	96	138	182
		5	87	150	216	285
Condition (3.10c)	M	2	21	40	64	93
		3	51	93	148	216
		4	93	170	269	390
		5	148	269	424	615

was proposed in [164] that under some approximations obtains generic uniqueness of $\mathbf{P}$ if

$$3\min(M, R) + \min(2M, R) \geq 2R + 3. \tag{3.14}$$

By inspection of Table 3.2, it is clear that Proposition 3.3.2 leads to more relaxed results, indicating that it exploits the problem structure better.

3.4 Algorithms for BSI

In this section we explain that a tensor decomposition with block-Hankel factor matrices such as (3.2) can be reduced to an unstructured third-order CPD problem by exploiting the subspace structure of the matrix $\mathbf{Y}$ defined in (3.7). This will lead to subspace methods using either the channel subspace or the noise subspace, which is defined as the subspace orthogonal to the channel subspace. These algorithms lead to the exact solution in the noiseless case when the conditions in Proposition 3.3.2 hold (plus an additional condition in the case of the noise subspace method). For simplicity, we will now consider the specific case where $\mathbf{H}$ is banded, as shown in Figure 3.1. The algorithms can be extended to general block-Hankel structured factors.

Table 3.2: Maximum value for R for which the conditions from [164, 131] and Proposition 3.3.2 generically guarantee uniqueness of the channel parameter matrix $\mathbf{P}$ in (3.4).

L	2					3					4					5				
M	2	3	4	5		2	3	4	5		2	3	4	5		2	3	4	5	
Condition (3.14) of [164]	3	6	8	11		3	6	8	11		3	6	8	11		3	6	8	11	
Condition (3.13) of [131]	6	9	12	15		8	12	16	20		10	15	20	25		12	18	24	30	
Proposition 3.3.2	12	27	48	75		13	29	52	81		13	30	54	85		14	31	56	87	

3.4.1 Channel Subspace Method

Consider the structured factor matrix $\mathbf{H}$ shown in Figure 3.1. From the structure of $\mathbf{H}$ it follows that by constructing the matrix $\mathbf{E}_{\mathbf{H}} \in \mathbb{C}^{M(2L+1)(L+1)\times M(L+1)}$ as

$$\mathbf{E}_{\mathbf{H}} = \begin{bmatrix} \mathbf{E}^{(1)} \\ \mathbf{E}^{(2)} \\ \vdots \\ \mathbf{E}^{(L+1)} \end{bmatrix}$$

$$\mathbf{E}^{(l)} = \begin{bmatrix} \mathbf{0}_{M(L+1-l)\times M(L+1)} \\ \mathbf{I}_{M(L+1)} \\ \mathbf{0}_{M(l-1)\times M(L+1)} \end{bmatrix},$$

it holds that

$$\mathbf{E}_{\mathbf{H}}\,\mathbf{P} = \begin{bmatrix} \mathbf{H}^{(0)} \\ \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(L)} \end{bmatrix}. \tag{3.15}$$

In this expression, $\mathbf{P}$ is given in equation (3.4) and the matrices $\mathbf{H}^{(l)} \in \mathbb{C}^{M(2L+1)\times R}$ are defined in equation (3.3).

Now define the matrix $\mathbf{Z}$ as

$$\mathbf{Z} = \mathbf{H}^* \odot \mathbf{H} \quad \in \mathbb{C}^{M^2(2L+1)^2\times R(L+1)},$$

Because of the Khatri–Rao structure of $\mathbf{Z}$ and the block-Hankel structure of $\mathbf{H}$, the counterpart of expression (3.15) is given by

$$\mathbf{E}\,(\mathbf{P}^* \odot \mathbf{P}) = \begin{bmatrix} \mathbf{Z}^{(0)} \\ \mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(L)} \end{bmatrix}, \tag{3.16}$$

in which $\mathbf{E} \in \mathbb{C}^{M^2(2L+1)^2(L+1)\times M^2(L+1)^2}$ is defined by

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}^{(1)} \otimes \mathbf{E}^{(1)} \\ \mathbf{E}^{(2)} \otimes \mathbf{E}^{(2)} \\ \vdots \\ \mathbf{E}^{(L+1)} \otimes \mathbf{E}^{(L+1)} \end{bmatrix}, \tag{3.17}$$

and the matrices $\mathbf{Z}^{(l)} \in \mathbb{C}^{M^2(2L+1)^2 \times R}$ are obtained from the partition

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}^{(0)} & \mathbf{Z}^{(1)} & \cdots & \mathbf{Z}^{(L)} \end{bmatrix}.$$

Let the column vectors of $\mathbf{U} \in \mathbb{C}^{M^2(2L+1)^2 \times R(L+1)}$ constitute a basis for range $(\mathbf{Z})$. This matrix can be constructed numerically by computing the dominant left singular vectors of the matricized tensor $\mathbf{Y}$ from (3.7), i.e., the left singular vectors corresponding to the largest $R(L+1)$ singular values. Note that we construct an orthogonal basis using the truncated singular value decomposition, which is an unstructured low-rank approximation (ULRA), but there are other ways to find a basis for $\mathbf{Z}$ (see e.g., [95]). Since the column spaces of $\mathbf{U}$ and $\mathbf{Z}$ are the same, there exists a nonsingular matrix $\mathbf{M} \in \mathbb{C}^{R(2L+1) \times R(2L+1)}$ such that

$$\mathbf{Z} = \mathbf{U}\mathbf{M}. \tag{3.18}$$

Note that if an ULRA is used, the choice of orthogonal basis $\mathbf{U}$ does not affect the solution since any other orthogonal basis only introduces a complex rotation which will be captured in $\mathbf{M}$. Let us partition $\mathbf{M}$ as follows

$$\mathbf{M} = [\mathbf{M}_1, \ldots, \mathbf{M}_{L+1}], \quad \mathbf{M}_l \in \mathbb{C}^{R(L+1) \times R},$$

then due to relation (3.18) we find that

$$\begin{bmatrix} \mathbf{Z}^{(0)} \\ \vdots \\ \mathbf{Z}^{(L)} \end{bmatrix} = (\mathbf{I}_{L+1} \otimes \mathbf{U}) \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{L+1} \end{bmatrix}.$$

By plugging in equation (3.16) we obtain

$$\mathbf{E}\,(\mathbf{P}^* \odot \mathbf{P}) = (\mathbf{I}_{L+1} \otimes \mathbf{U}) \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{L+1} \end{bmatrix}, \tag{3.19}$$

which is the R-column extension of the equation obtained in [115]. Equation (3.19) is equivalent to

$$[\mathbf{I}_{L+1} \otimes \mathbf{U}, -\mathbf{E}] \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{L+1} \\ \mathbf{P}^* \odot \mathbf{P} \end{bmatrix} = \mathbf{0}. \tag{3.20}$$

Since the matrices $\mathbf{U}$ and $\mathbf{E}$ are known, extracting $\mathbf{P}$ is equivalent to solving this linear system under a Khatri–Rao constraint on the last part of the solution.

We can relax this problem by first solving the system without imposing any structure. If the conditions stated in Lemma 3.3.1 are satisfied, then we know that the unstructured solution to (3.20) is unique up to right multiplication by a nonsingular matrix $\mathbf{A} \in \mathbb{C}^{R \times R}$. This means that the unstructured solution to this linear system will have the same column space as the structured solution. Let $\mathbf{Q} \in \mathbb{C}^{(R+M^2)(L+1)^2 \times R}$ denote a matrix of which the columns form a basis for the kernel of $[\mathbf{I}_{L+1} \otimes \mathbf{U}, -\mathbf{E}]$. Finding $\mathbf{Q}$ can be done numerically by computing the right singular vectors of $[\mathbf{I}_{L+1} \otimes \mathbf{U}, -\mathbf{E}]$ associated with the smallest R singular values. Partition $\mathbf{Q}$ into

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix},$$

such that $\mathbf{Q}_2 \in \mathbb{C}^{M^2(L+1)^2 \times R}$ has the same dimensions as $\mathbf{P}^* \odot \mathbf{P}$. It then remains to solve

$$\mathbf{Q}_2 = (\mathbf{P}^* \odot \mathbf{P})\, \mathbf{A}^{\mathrm{T}}. \tag{3.21}$$

We recognize an unfolded CPD in equation (3.21). Extracting $\mathbf{P}$ from this equation is equivalent to computing the CPD of the tensorized matrix $\mathbf{Q}_2$. If condition (3.10c) in Proposition 3.3.2 is satisfied, this decomposition is unique. In the exact case, the solution can be found using a generalized eigenvalue decomposition [51, 87]. However, any algorithm can be used to solve this CPD. In the noisy case, this CPD will be approximate.

Note that the convolutive shift structure is no longer present in equation (3.21). What remains is a linear instantaneous mixture of the system coefficients. Consequently, the procedure up to equation (3.21) can be seen as a reduction from a convolutive to an instantaneous mixture. An overview of the channel subspace algorithm is given in Algorithm 1.

The complexity of the channel subspace algorithm using a GEVD in the final step is in the order of $\mathcal{O}\left(24M^2R^2L^4 + 11R^3L^3 + 16M^6L^7\right)$. In this expression, the first two terms stem from the computation of $\mathbf{U}$ and the third term is due to the computation of $\mathbf{Q}$.

3.4.2 Noise Subspace Method

While the previous channel subspace method considered the column space of $\mathbf{H}^* \odot \mathbf{H}$, the noise subspace method looks at the space orthogonal to this to solve the BSI problem. A necessary condition for this method is that $M^2(2L+1)^2 > R(L+1)$.

Algorithm 1: Overview of the channel subspace algorithm. See text for numerical aspects.

Data: (Augmented) matricized tensor $\mathbf{Y} \in \mathbb{C}^{M^2(2L+1)^2 \times 2M^2(2L+1)}$ of the form (3.7)

Result: Estimate of system coefficients $\widehat{\mathbf{P}} \in \mathbb{C}^{M(L+1)\times R}$

1. Reduce to an instantaneous mixture
   - Compute $\mathbf{U} \in \mathbb{C}^{M^2(2L+1)^2 \times R(L+1)}$ so that $\text{range}\,(\mathbf{U}) = \text{range}\,(\mathbf{Y})$;
   - Use (3.17) to construct $[\mathbf{I}_{L+1} \otimes \mathbf{U}, -\mathbf{E}] \in \mathbb{C}^{M^2(2L+1)^2(L+1)\times(R+M^2)(L+1)^2}$;
   - Find $\mathbf{Q} \in \mathbb{C}^{(R+M^2)(L+1)^2 \times R}$ of which the columns form a basis for the kernel of $[\mathbf{I}_{L+1} \otimes \mathbf{U}, -\mathbf{E}]$;
2. Identify the system
   - Compute the (approximate) CPD in equation (3.21);

Assume that $\mathbf{H}^* \odot \mathbf{H}$ has full column rank, then

$$\dim\left(\text{range}\,(\mathbf{H}^* \odot \mathbf{H})^{\perp}\right) = M^2(2L+1)^2 - R(L+1) =: Q.$$

Let the column vectors of $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_Q] \in \mathbb{C}^{M^2(2L+1)^2 \times Q}$ constitute a basis for $\text{range}\,(\mathbf{H}^* \odot \mathbf{H})^{\perp}$. Such a basis can be found by computing the left singular vectors of the matricized tensor $\mathbf{Y}$ from (3.7) corresponding to the Q smallest singular values.

Due to the fact that $\mathbf{v}_i \in \text{range}\,(\mathbf{H}^* \odot \mathbf{H})^{\perp}$ we have

$$\mathbf{V}^H (\mathbf{H}^* \odot \mathbf{H}) = \mathbf{0}.$$

We relax the Khatri–Rao structure and instead solve

$$\mathbf{V}^H \mathbf{B} = \mathbf{0}, \tag{3.22}$$

with $\mathbf{B}$ a matrix of full column rank having the same banded block-Hankel structure as $\mathbf{H}^* \odot \mathbf{H}$. Note that because $\mathbf{V}$ forms a basis for $\text{range}\,(\mathbf{H}^* \odot \mathbf{H})^{\perp}$, it holds that $\text{range}\,(\mathbf{B}) = \text{range}\,(\mathbf{H}^* \odot \mathbf{H})$. Using the matrix $\mathbf{E}$ from equation (3.17) again, we can write

$$\mathbf{EQ} = \begin{bmatrix} \mathbf{B}^{(0)} \\ \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{B}^{(L)} \end{bmatrix}, \tag{3.23}$$

in which $\mathbf{Q} \in \mathbb{C}^{M^2(L+1)^2 \times R}$ is the generating block of $\mathbf{B}$ and the matrices $\mathbf{B}^{(l)} \in \mathbb{C}^{M^2(2L+1)^2 \times R}$ are obtained from the partition

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{(0)} & \mathbf{B}^{(1)} & \cdots & \mathbf{B}^{(L)} \end{bmatrix}.$$

Using (3.23), we can rewrite equation (3.22) as

$$\mathbf{W}\mathbf{Q} = \mathbf{0}, \tag{3.24}$$

with $\mathbf{W}$ constructed as

$$\mathbf{W} = \left(\mathbf{I}_{L+1} \otimes \mathbf{V}^{\mathrm{H}}\right) \mathbf{E}. \tag{3.25}$$

Finding a nontrivial solution to (3.24) can be done numerically by computing the right singular vectors of $\mathbf{W}$ associated with the R smallest singular values. Any nontrivial solution to (3.24) will be unique up to right multiplication by a nonsingular matrix $\mathbf{A} \in \mathbb{C}^{R \times R}$. More specifically, we know that $\mathbf{B}$ has the same block-Hankel structure as $\mathbf{H}^* \odot \mathbf{H}$ and that $\operatorname{range}(\mathbf{B}) = \operatorname{range}(\mathbf{H}^* \odot \mathbf{H})$. Assuming $\mathbf{D}$ and $\mathbf{N}^* \odot \mathbf{N}$ have full column rank, it then follows from Lemma 3.3.1 that $\mathbf{B}(\mathbf{I}_{L+1} \otimes \mathbf{A}) = \mathbf{H}^* \odot \mathbf{H}$, with $\mathbf{A} \in \mathbb{C}^{R \times R}$ a nonsingular matrix. From the block-Hankel structure it then follows that $\mathbf{Q}\mathbf{A} = \mathbf{P}^* \odot \mathbf{P}$. Note that $\mathbf{Q}$ and $\mathbf{P}^* \odot \mathbf{P}$ are the generating blocks of $\mathbf{B}$ and $\mathbf{H}^* \odot \mathbf{H}$, respectively. Consequently, for any nontrivial solution to (3.24) we can write

$$\mathbf{Q} = (\mathbf{P}^* \odot \mathbf{P})\, \mathbf{A}^{-1}, \tag{3.26}$$

from which $\mathbf{P}$ can be computed in a similar way as in the previous section. An overview of the noise subspace algorithm is given in Algorithm 2.

The complexity of the noise subspace algorithm using a GEVD in the final step is in the order of $\mathcal{O}\left(256M^6L^7\right)$, which is mainly due to the computation of $\mathbf{Q}$.

3.5 Numerical experiments

As a proof of concept, the developed algorithms will first be applied to some synthetic tensors. In a next experiment, they will be applied to blind system identification.

3.5.1 Structured tensor decomposition

Consider a tensor $\mathcal{A} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \in \mathbb{C}^{15 \times 15 \times 25}$. Both $\mathbf{A}$ and $\mathbf{B}$ have size 15×21 and are entirely fixed by submatrices $\mathbf{P}_{\mathbf{A}}$ and $\mathbf{P}_{\mathbf{B}}$ of size 9×7, which are

Algorithm 2: Overview of the noise subspace algorithm. See text for numerical aspects.

Data: (Augmented) matricized Tensor $\mathbf{Y} \in \mathbb{C}^{M^2(2L+1)^2 \times 2M^2(2L+1)}$ of the form (3.7)

Result: Estimate of system coefficients $\widehat{\mathbf{P}} \in \mathbb{C}^{M(L+1)\times R}$

1. Reduce to an instantaneous mixture
   - Compute $\mathbf{V} \in \mathbb{C}^{M^2(2L+1)^2 \times M^2(2L+1)^2 - R(L+1)}$ so that $\operatorname{range}(\mathbf{V}) = \operatorname{range}(\mathbf{Y})^{\perp}$;
   - Use (3.25) to construct $\mathbf{W} \in \mathbb{C}^{M^2(2L+1)^2(L+1) - R(L+1)^2 \times M^2(L+1)^2}$;
   - Find $\mathbf{Q} \in \mathbb{C}^{M^2(L+1)^2 \times R}$ of which the columns form a basis for the kernel of $\mathbf{W}$;
2. Identify the system
   - Compute the (approximate) CPD in equation (3.26);

repeated in a banded block-Hankel structure just as the matrix $\mathbf{P}$ depicted in Figure 3.1. For clarity, we mention that this corresponds to $M = 3$, $R = 7$ and $L = 2$. In an identification context, this would be an underdetermined system with 7 inputs and only 3 outputs. The entries of $\mathbf{P_A}$ and $\mathbf{P_B}$, and the entries of the third factor matrix $\mathbf{C} \in \mathbb{C}^{25\times 21}$ have real and imaginary parts that are randomly drawn from a standard normal distribution. In the experiment, Gaussian noise was added to the constructed tensor to obtain various signal-to-noise ratios (SNR). The noisy tensor $\mathcal{A}$ is decomposed using the channel and the noise subspace-based methods presented in this paper. For both methods, the CPD in the second step of the algorithm, given by equation (3.21) or (3.26), is computed once using a generalized eigenvalue decomposition (GEVD) and once using a nonlinear least squares algorithm (NLS). The error on the result equals the mean of the relative errors on $\mathbf{P_A}$ and $\mathbf{P_B}$, taking the scaling and permutation ambiguities into account. Mathematically, the relative error norm (REN) can be written as

$$\begin{aligned}&\operatorname{REN}(\mathbf{P_A}, \mathbf{P_B})\\ &= 20\log_{10}\left(\frac{\operatorname{RelErr}(\mathbf{P_A}) + \operatorname{RelErr}(\mathbf{P_B})}{2}\right)[\mathrm{dB}],\end{aligned}$$

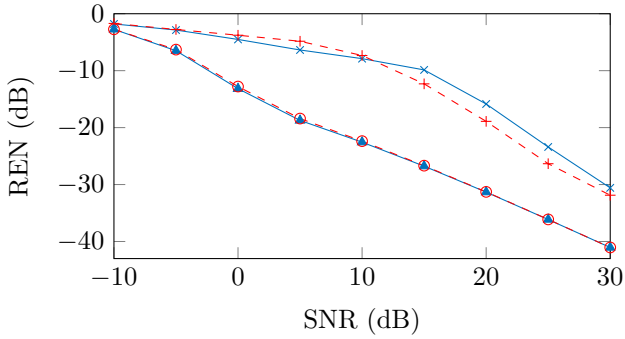


Figure 3.3: Comparison of the median performance of the channel subspace method with a GEVD solution (—×—) or a NLS solution (—▲—) to (3.21), and the noise subspace method with a GEVD solution (- +-) or a NLS solution (- ⊙-) to (3.26) for a (noisy) tensor with two different banded block-Hankel factors.

with

$$\text{RelErr}(\mathbf{P_N}) = \frac{\left|\left|\mathbf{P_N} - \widehat{\mathbf{P}}_\mathbf{N}\Delta_{\text{opt,n}}\Pi_{\text{opt,n}}\right|\right|_\text{F}}{||\mathbf{P_N}||_\text{F}},$$

in which $\Delta_{\text{opt,n}}$ and $\Pi_{\text{opt,n}}$ represent the matrices for optimal scaling and permutation, respectively. These ambiguity matrices are determined using the `cpderr` command in TENSORLAB [156]. The median result of 200 experiments for each SNR value is shown in Figure 3.3. It can be seen that both the channel and noise subspace algorithm perform better when less noise is present, as expected, and they give very good estimates at high SNRs. The figure also shows that computing the CPD from (3.21) or (3.26) using a NLS algorithm yields significant improvements over a GEVD computation at the cost of a higher computational complexity. It can be verified that when no noise is present, all depicted algorithms find the solution up to machine precision.

3.5.2 Application to blind system identification

Consider a real linear FIR system with $M = 5$ outputs, $R = 4$ inputs and $L = 2$. The inputs consist of 10^5 i.i.d. samples randomly chosen from a uniform distribution on $[-\frac{1}{2}, \frac{1}{2}]$. The system coefficients are randomly drawn from a standard normal distribution. The additive noise in the experiment is also randomly chosen from a standard normal distribution and is subsequently scaled to obtain the desired SNR value.

As explained, blind identification of the system amounts to the computation of a Hankel-structured decomposition of a tensor $\mathcal{T} = [\![\mathbf{G}, \mathbf{H}, \mathbf{H}, \mathbf{H}]\!]$ with $\mathbf{H}$ as depicted in Figure 3.1. The performance is evaluated by means of the relative error norm, which reduces to

$$\mathrm{REN}(\mathbf{P}) = 20 \log_{10} \left(\frac{\left|\left| \mathbf{P} - \widehat{\mathbf{P}} \Delta_{\mathrm{opt}} \Pi_{\mathrm{opt}} \right|\right|_{\mathrm{F}}}{||\mathbf{P}||_{\mathrm{F}}} \right) [\mathrm{dB}] \,.$$

As in the previous experiment, Δ_{opt} and Π_{opt} represent the matrices for optimal scaling and permutation, which are determined using the `cpderr` command in TENSORLAB [156]. All reported values are the mean of 200 experiments.

First, we will compare our subspace methods with a direct decomposition approach and with another subspace-based approach in the literature. Next, we will illustrate the use of our presented method as an initialization method for optimization-based methods that take all structure into account.

Comparison of methods

We compare the accuracy and time complexity of our channel and noise subspace methods with alternative methods. We compare methods that decompose the same structured cumulant tensor $\mathcal{T}$, though we want to mention that other techniques for BSI exist, which for instance deconvolve the outputs before computing statistics [165, 139, 91]. A comparison of all approaches is out of the scope of this paper.

One straightforward way to find the system coefficients is to directly decompose the full cumulant tensor $\mathcal{T}$ in rank-1 terms and extract the system coefficients using clustering. More specifically, because the CPD factors are only determined up to scaling and permutation, we have to permute the columns of the estimated factors such that the block-Hankel structured emerges. Only then it is possible to extract the system coefficients from one of the block-columns of $\widehat{\mathbf{H}}$. Since we only need one block column, e.g., the first one, we can cluster the columns of the factor matrix by selecting the R columns of which the first ML entries have the smallest norm. Note from the structure of $\mathbf{H}$ presented in Figure 3.1 that these entries are zero in the exact case. From these vectors, $\widehat{\mathbf{P}}$ can be simply extracted. The decomposition of the full cumulant tensor is computed using a GEVD or an NLS method. We will refer to this method as the direct decomposition approach (with GEVD or with NLS).

Another method has been presented by Liang in [88]. It is a subspace-based method that exploits the block-Hankel structure in one of the factor matrices.

We apply this method to our cumulant tensor $\mathbf{T}$ instead of the set of cumulant matrices proposed in [88] to allow fair comparisons, but the idea remains the same. After exploiting the structure, Liang proposes to resolve the remaining ambiguities using either a matrix pencil based approach or a CPD. We will use the latter and will compute this CPD with both a GEVD and an NLS method.

The accuracy of the different methods for various SNR values is shown in Figure 3.4. It can be seen that all methods perform poorly at low SNR values. This is due to perturbations by added noise and the estimation error of the cumulant as explained at the end of Section 3.2. This also implies that time-domain blind system identification in high noise regime is a very difficult problem. As the SNR increases, the noise influence lessens, but the cumulant estimation error remains.

It is clear from Figure 3.4 that the direct decomposition method using a GEVD performs the worst. This is because the method does not take any structure into account and because it uses a GEVD, which is only optimal in the noiseless case. The figure also shows that our NLS-based subspace methods outperform all other methods. Following the same reasoning as before, this is because the structure in two modes is exploited and an NLS-type method is used in the second step of the algorithm, as discussed in Section 3.5.1. The other methods yield comparable results. Note that for each method, the variant using NLS outperforms the variant using a GEVD. For high SNRs, exploiting structure also leads to better results. For instance, the NLS variant of Liang's method outperforms the direct decomposition but is worse than the subspace method presented in this paper, which exploits the structure in two modes.

Apart from accuracy, we can also compare the computation times of the different methods. The mean results over 200 experiments at an SNR of 15 dB are given in Table 3.3. These timings were obtained on a standard laptop and do not include the computation of the higher-order statistics. The table shows that our subspace methods are among the fastest algorithms of all algorithms tested. Only the direct decomposition using the GEVD can compete in terms of time complexity, but does not come close in terms of accuracy.

Subspace methods as initialization

We show that the results of our subspace-based methods can also be used as initialization for optimization-based methods. The two optimization-based algorithms considered here are the single-step least squares algorithm as presented in [59] and a fully structured tensor decomposition implemented in Tensorlab [114, 156]. Both take the problem structure fully into account.

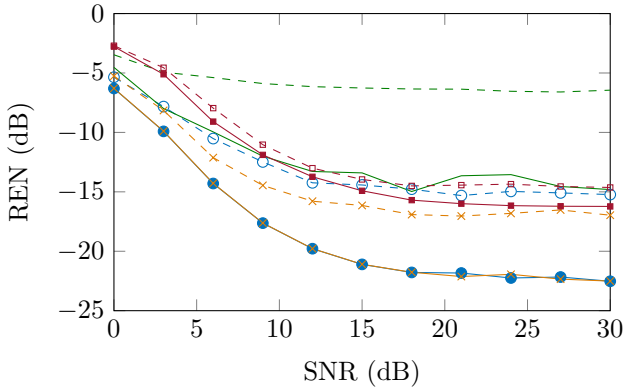


Figure 3.4: Comparison of the mean performance for a blind system identification problem with $M = 5$, $R = 4$ and $L = 2$. The figure shows the channel subspace method with GEVD (- ⊙ -) and with NLS (—●—), the noise subspace method with GEVD (- × -) and with NLS (—×—), the direct decomposition approach with GEVD (- - -) and with NLS (——), the approach from [88] using GEVD (- ▫ -) and using NLS (—▪—).

Table 3.3: Average computation time over 200 experiments for various BSI methods computed on a standard laptop. The considered system has parameters $M = 5$, $R = 4$ and $L = 2$ and outputs of length 10^5 with 15 dB SNR. The table only shows the decomposition time, i.e., the time needed to compute the higher-order statistics is not included.

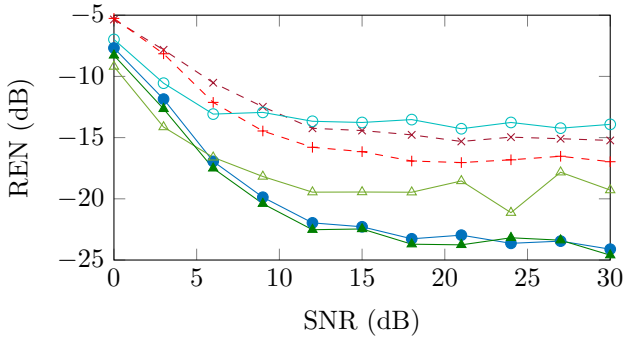
Method	Time
Channel subspace method with GEVD	0.055 s
Channel subspace method with NLS	0.145 s
Noise subspace method with GEVD	0.069 s
Noise subspace method with NLS	0.154 s
Direct decomposition approach with GEVD	0.085 s
Direct decomposition approach with NLS	2.727 s
Liang's approach [88] with GEVD	0.666 s
Liang's approach [88] with NLS	0.861 s
Tensorlab [156] implementation with random initialization	1.741 s
Tensorlab [156] impl. with noise subspace (NLS) initialization	1.148 s
Method from [59] with random initialization	0.379 s
Method from [59] with noise subspace (NLS) initialization	0.368 s

As opposed to subspace-based methods, optimization-based methods avoid the use of the perturbed subspace by directly comparing the decomposition to the (noisy) data tensor. Because of this, one expects better accuracy than the subspace-based methods since errors cannot accumulate throughout different steps of the algorithm. However, Figure 3.5a shows that the randomly initialized optimization-based method implemented in TENSORLAB performs rather poorly. This is because the random initialization is often not a good starting point and the algorithm gets stuck in a local minimum. The figure also shows that the method from [59] is more robust to bad initializations. Rerunning the algorithm is possible, but is time consuming and is no guarantee to a better solution. In this experiment, the randomly initialized optimization-based algorithms were run 4 times in each experiment to allow several initializations. Of these 4 runs, the best result is retained.

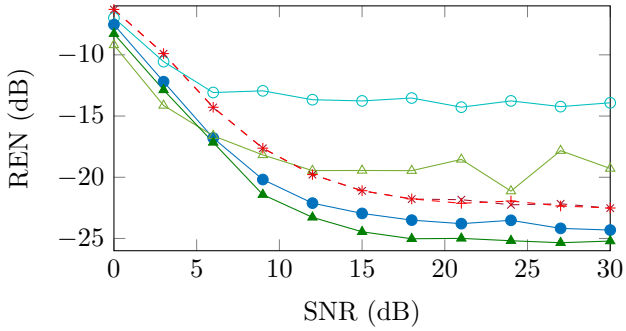
A more sensible approach to initialize the optimization-based algorithms consists of using the developed subspace-based methods, which offer a reasonably accurate initial value. Figure 3.5 illustrates this approach for the optimization-based method implemented in TENSORLAB and the method presented in [59]. The figure indeed shows that this two-step approach outperforms the others when the subspace-based initialization is good. Using a GEVD or NLS-type method in the second step of the noise subspace initialization method yields comparable results. In terms of computation time, our subspace-based methods were shown to be fast in the previous experiment and consequently provide a cheap initialization. Moreover, the optimization-based methods will need fewer iterations to converge when properly initialized and do not require multiple reruns. This is illustrated in the bottom part of Table 3.3: both optimization-based methods are faster when they are properly initialized (note that the initialization method is also included in these computation times).

Impact of longer filters

To illustrate the effects of longer FIR filters, numerical simulations are done for a system with $M = 5$ outputs and $R = 3$ inputs for various channel lengths. The signal to noise ratio is fixed at 20 dB and 10^5 observations are used to compute the fourth-order statistics. The source signals and channel coefficients are generated randomly in the same way as in the previous experiments. Figure 3.6 shows the mean accuracy and time complexity of both the channel and noise subspace methods over 20 repetitions at each value for L. It is clear from the figure that the accuracy drops and the computation time rises significantly when increasing the maximum filter delay L. This is not surprising as the number of entries of the tensor that needs to be decomposed increases by L^3 and more parameters must be estimated using the same amount of output

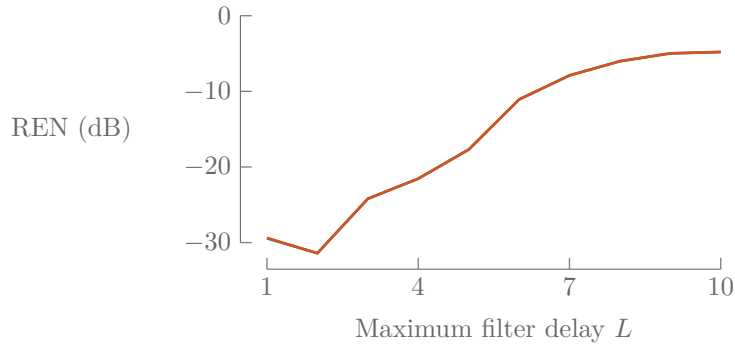


(a) Performance when the subspace methods use a GEVD in the second step of the algorithm.

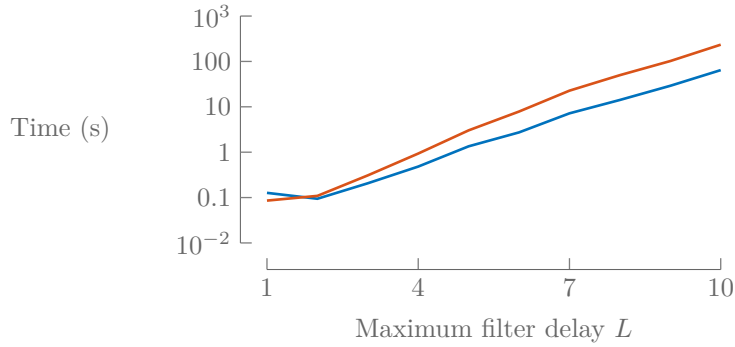


(b) Performance when the subspace methods use a NLS approach in the second step of the algorithm..

Figure 3.5: Comparison of the mean performance for a blind system identification problem with $M = 5$, $R = 4$ and $L = 2$. The figure shows the channel subspace method (- × -), noise subspace method (- + -) and optimization-based methods initialized in different ways. More specifically, a TENSORLAB implementation is shown with random initialization (—○—) and initialized with the noise subspace method (—●—), and the method from [59] is shown with random initialization (—△—) and initialized with the noise subspace method (—▲—).



(a) Accuracy (expressed as relative error norm) as a function of the maximum filter delay of the FIR system. The values for the channel and noise subspace methods differ only slightly, hence only one is visible in the graph.



(b) Computation timeas a function of the maximum filter delay of the FIR system.

Figure 3.6: Mean accuracy and time complexity of both the channel (blue) and noise (red) subspace methods when the maximum filter delay L is varied.

information. We also mention at this point that increasing the number of inputs R while keeping the filter lengths and number of outputs constant will reduce the accuracy of the method for the same reason, i.e. more parameters need to be estimated based on the same amount of information.

3.6 Conclusions

A new deterministic and generic uniqueness condition for tensor decompositions with at least two block-Hankel structured factor matrices was introduced by exploiting the available structure. This condition is a significant improvement over existing results. Using subspaces, two algebraic algorithms were devised reducing a banded Hankel-structured tensor decomposition to a third-order unstructured CPD.

This theory and the algorithms can be applied to linear FIR blind system identification of systems with i.i.d. inputs, which may even be underdetermined. It has been shown that the subspace-based methods offer a cheap and accurate initialization for structured optimization-based methods through numerical experiments.

Chapter 4

Coupled and incomplete tensors in blind system identification

Abstract — Blind system identification (BSI) is an important problem in signal processing, arising in applications such as wireless telecommunications, biomedical signal processing and seismic signal processing. In the past decades, tensors have proven to be useful tools for these blind identification and separation problems. Most often, tensor-based methods based on fourth-order statistics are used, which have been studied extensively for independent component analysis and its convolutive extensions. However, these tensor-based methods have two main drawbacks: the accuracy is often limited by the estimation error of the statistics and the computation of these fourth-order statistics is time-intensive. In this paper, we propose to counter these drawbacks for BSI by coupling the fourth-order statistics with second-order statistics and by using incomplete tensors. By doing so, we can obtain more accurate results or obtain results in a much faster way.

Reference — This chapter is a slightly adapted version of [146]. The candidate did the research and wrote the article under the guidance of the coauthor.

4.1 Introduction

Traditional system identification uses input and output information to compose a system description. In some applications, however, the input signals are expensive or even impossible to measure. For instance, the inputs of a transmission system in telecommunications are the signals of interest and are not known a priori. However, we still wish to have a system description to counter possible inter symbol interference [2]. This system description is often estimated using known pilot sequences, which take up valuable bandwidth. Alternatively, the system can be estimated in a blind way, which is more challenging to do accurately but requires less bandwidth. Other examples of blind system identification can be found in biomedical signal processing, speech separation and seismic signal processing [2, 73, 103, 28]. In these applications, one can resort to blind system identification (BSI), which estimates a system using solely the output signals. To make the BSI problem feasible, some additional assumptions on the input signals or mixing coefficients are needed. Popular assumptions include statistically independent inputs [28, 23], low-rank mixing vectors [14], finite alphabet [138], constant modulus [137] and sparse inputs [166].

In this paper, we consider linear time-invariant finite impulse response (FIR) systems with independent and identically distributed (i.i.d.) inputs. This model can for instance be used in telecommunications, where the time delays of the FIR system model path length differences. Note that this problem statement is similar to convolutive independent component analysis (ICA), with the conceptual difference that ICA focuses on retrieving the inputs, whereas BSI aims to find a system description [28]. We will use both terms throughout this text, but will focus on retrieving the system description rather than the inputs.

Tensors are higher-order generalizations of vectors and matrices. One key strength of tensors is that their decompositions are unique under mild conditions, which is not the case for matrices [31, 84, 52, 110]. Moreover, many decompositions can be computed algebraically, which leads to fast algorithms for reasonably small tensors when compared to optimization-based approaches [51, 123]. Because of these properties, tensors are well-established for the identification of instantaneous mixtures of independent components [28, 110, 81, 25] and are increasingly surfacing for convolutive ICA, see e.g. [164, 59, 12]. Independent components and tensors are naturally linked through higher-order moments and cumulants. For instance, a fourth-order cumulant of a stochastic vector immediately yields a fourth-order tensor. If the elements of the stochastic vector are statistically independent, the resulting tensor is diagonal.

The existing tensor-based methods for BSI with independent inputs follow various approaches. One common approach is to reduce the convolutive mixture to an instantaneous one, which can then be identified using existing (tensor-based) techniques. The deconvolution can be done by transforming the data to the frequency domain (see [103] and references therein) or by deconvolution techniques in the time domain (e.g., [138, 61, 94, 91, 139, 141]). Another approach immediately tensorizes the data and subsequently decomposes the resulting tensor [164, 59, 12, 147, 88].

In this paper, we focus on tensor-based methods in the time domain that use fourth-order statistics to identify the system. These methods typically have two downsides. First, the accuracy of the estimated statistics may be low if the signals are short. This drawback can be alleviated by combining both second- and fourth-order statistics. Traditionally, this is often done through prewhitening, but this does not allow the user to weigh the importance of the fourth-order statistics and neglects possible noise influences on the second-order statistics [40, 21]. As a first contribution, we will counter these disadvantages for BSI by combining second- and fourth-order statistics using coupled tensor decompositions [123, 5, 85, 114, 117]. Another drawback of tensor-based methods using fourth-order statistics is the computation time of constructing the cumulant tensor. Even though fourth-order cumulants are cheaper than, e.g., negentropy [71], computing a fourth-order cumulant can be expensive for long signals and systems with many outputs.

As a second contribution, we propose to improve the time complexity by constructing just a subset of the statistics, which yields an incomplete tensor. This incomplete tensor can then be decomposed using existing techniques [114, 129, 3, 76, 155, 152, 120]. This approach allows us to exchange a slight loss in accuracy for a drastically improved time complexity.

The rest of this section introduces the notations used throughout this text. Next, tensors and their decompositions are presented, after which the problem at hand is introduced in Section 4.3. Coupled and incomplete statistics are subsequently treated in Sections 4.4 and 4.5. Finally, the results are illustrated in Section 4.6 using numerical experiments.

Notations

Scalars are denoted by lowercase letters (e.g., a), vectors by bold lowercase letters (e.g., $\mathbf{a}$), matrices by bold uppercase letters (e.g., $\mathbf{A}$), and tensors by uppercase calligraphic letters (e.g., $\mathcal{A}$). The outer product is denoted by $\circ$. The Kronecker product is denoted by $\otimes$ and the Khatri–Rao product is given by $\odot$. The complex conjugate is denoted by $\cdot^*$ and the conjugate transpose

(Hermitian transpose) by $\cdot^{\mathrm{H}}$. The (cross)cumulant of signals $a(n), b(n), c(n), d(n)$ is denoted by $\mathrm{Cum}[a(n), b(n), c(n), d(n)]$. The Frobenius norm is given by $||\cdot||_{\mathrm{F}}$. Tensor-matrix products in the nth mode are denoted by $\cdot_n$ [81]. Estimates are denoted by a hat atop, e.g., $\widehat{a}$.

4.2 Canonical polyadic decomposition

We say that an Nth-order tensor has rank 1 if and only if it equals the outer product of N nonzero vectors. By extension, the rank of a tensor $\mathcal{A}$ is defined as the minimal number of rank-1 tensors yielding $\mathcal{A}$ in a linear combination.

The polyadic decomposition (PD) writes a tensor $\mathcal{A}$ as a linear combination of rank-1 terms:

$$\begin{aligned}\mathcal{A} &= \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(N)} \\ &= \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right],\end{aligned} \tag{4.1}$$

in which $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_R]$ and the factor matrices $\mathbf{U}^{(n)}$ are defined as $\mathbf{U}^{(n)} = \left[\mathbf{u}_1^{(n)}, \cdots, \mathbf{u}_R^{(n)}\right]$ for $n \in \{1, \ldots, N\}$. If $\boldsymbol{\lambda}$ is a vector containing only ones, it can be omitted from the notations. Another notation for this PD is

$$\mathcal{A} = \mathcal{L} \cdot_1 \mathbf{U}^{(1)} \cdot_2 \cdots \cdot_N \mathbf{U}^{(N)},$$

in which $\mathcal{L}$ is a core tensor of size $R \times R \times \cdots \times R$ consisting of all zero entries except for the diagonal, which contains the values λ_1 to λ_R. If R in (4.1) is minimal, $\mathcal{A}$ has rank R and we call the decomposition canonical (canonical polyadic decomposition, abbreviated as CPD). For historical purposes we mention that CANDECOMP and PARAFAC are other terms sometimes used for this decomposition [110, 81, 25].

Contrary to a decomposition of a matrix in rank-1 terms, the CPD of higher-order tensors is essentially unique under fairly mild conditions [84, 108, 50, 110]. Essential uniqueness means that the decomposition is unique up to trivial indeterminacies. More specifically, the different rank-1 terms can be permuted arbitrarily and the vectors within each rank-1 term may be scaled and counterscaled as long as the rank-1 term remains unchanged.

4.3 Problem statement

4.3.1 Blind system identification

Consider an FIR system with R inputs, M outputs and filter length $L+1$ per input-output channel. The mth output at the nth sample period can then be written as

$$y_m(n) = \sum_{r=1}^{R} \sum_{l=0}^{L} h_{mr}(l) s_r(n-l) + v(n), \tag{4.2}$$

in which $s_r(n)$ are the unknown input signals, $h_{mr}(n)$ are the unknown channel coefficients and $v(n)$ represents additive noise.

The goal of BSI is to retrieve the channel coefficients $h_{mr}(l)$ using solely the output signals. To do this, we make following assumptions:

1. The input signals $s_r(n)$ are zero-mean, non-Gaussian i.i.d. processes. The input signals are mutually statistically independent. This is a natural assumption in telecommunications applications, for example.
2. The additive Gaussian i.i.d. noise $v_m(n)$ has zero mean, is independent of the input signals and has the same variance for each output.

The input signals, mixing coefficients and the additive noise can be either real or complex-valued. Below, we derive the methods using complex signals.

Many tensor-based time-domain approaches to BSI use fourth-order spatio-temporal cumulants, which depend on four "spatial" parameters $m_x \in \{1, \ldots, M\}$ and three temporal parameters $\tau_x \in \{-L, \ldots, L\}$. This is motivated by the fact that fourth-order cumulants are blind to additive Gaussian noise. Note that this implies that if the noise is non-Gaussian, the resulting cumulant tensor will be perturbed by noise, which will either have to be explicitly extracted or not taken into account in order to avoid influencing the resulting estimates. Mathematically, an entry of the spatio-temporal cumulant is given by

$$\begin{aligned} &c_{m_1,m_2,m_3,m_4}(\tau_1, \tau_2, \tau_3) \\ &= \mathrm{Cum}[y^*_{m_1}(n), y_{m_2}(n+\tau_1), y^*_{m_3}(n+\tau_2), y_{m_4}(n+\tau_3)]. \end{aligned} \tag{4.3}$$

As explained in [59, 147], these statistics can be rearranged in a fourth-order tensor $\mathcal{T} \in \mathbb{C}^{M \times M(2L+1) \times M(2L+1) \times M(2L+1)}$ admitting a rank-$R(L+1)$ CPD under our two assumptions. More specifically, $\mathcal{T}$ can be written as

$$\mathcal{T} = \sum_{r=1}^{R} \gamma_r \sum_{l=0}^{L} \mathbf{p}_r^{(l)*} \circ \mathbf{h}_r^{(l)} \circ \mathbf{h}_r^{(l)*} \circ \mathbf{h}_r^{(l)}, \tag{4.4}$$

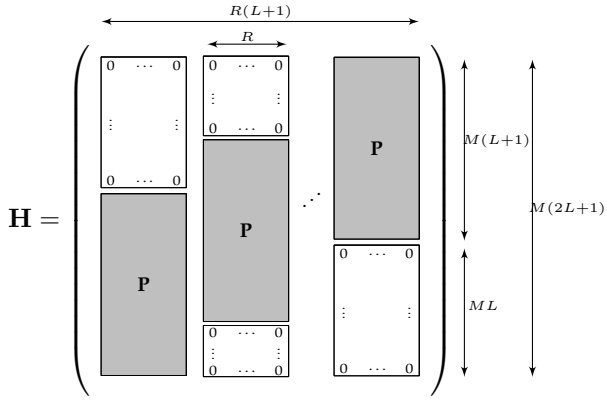


Figure 4.1: Block-Hankel structure of the factor matrix $\mathbf{H}$. This figure is adopted from [147] and slightly altered.

in which γ_r is the kurtosis of the rth input signal $s_r(n)$ and $\mathbf{p}_r^{(l)}$ is the rth column of $\mathbf{P}^{(l)}$, defined as

$$\mathbf{P}^{(l)} = \begin{bmatrix} h_{11}(l) & \cdots & h_{1R}(l) \\ \vdots & \ddots & \vdots \\ h_{M1}(l) & \cdots & h_{MR}(l) \end{bmatrix} \in \mathbb{C}^{M \times R},$$

for $l \in \{0, \ldots, L\}$. The vector $\mathbf{h}_r^{(l)}$ denotes the rth column of $\mathbf{H}^{(l)}$, which is given by

$$\mathbf{H}^{(l)} = \begin{bmatrix} \mathbf{0}_{M(L-l),R} \\ \mathbf{P} \\ \mathbf{0}_{Ml,R} \end{bmatrix} \in \mathbb{C}^{M(2L+1) \times R},$$

with $l \in \{0, \ldots, L\}$. The matrix $\mathbf{P}$ is defined by

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{(0)} \\ \vdots \\ \mathbf{P}^{(L)} \end{bmatrix} \in \mathbb{C}^{M(L+1) \times R}, \tag{4.5}$$

and contains all system parameters. More precisely, the rth column of $\mathbf{P}$ contains the parameters related to the rth system input. A graphical representation of the factor matrix $\mathbf{H} = \left[\mathbf{H}^{(0)}, \ldots, \mathbf{H}^{(L)}\right]$ is given in Figure 4.1. Note that this matrix is highly structured. When decomposing $\mathcal{T}$, taking the structure of $\mathbf{H}$ into account may yield better results, but is more complex as well.

Since the goal of blind system identification is to find all system coefficients, we wish to retrieve the matrix $\mathbf{P}$.

Note: throughout the paper we assume the number of inputs R is known. In practice, this is not always the case. Computing the rank of a tensor is an NP-hard problem [67, 66], but in some situations it is possible to get an idea of the number of inputs. If the system is strictly overdetermined, we can rewrite finding R as a matrix rank estimation problem as outlined in [138]. If extra information on the input signals is available, for instance if a finite alphabet is used, one could try several values for R in the algorithms and check whether the inputs behave as expected.

4.3.2 Challenges

Though constructing $\mathcal{T}$ and computing its (structured) decomposition in (4.4) is possible, there are a few challenges which we shall address here. First, fourth-order statistics are hard to estimate accurately. Second, computing these statistics may be computationally expensive. Below, we elaborate on these issues and give an overview of ways to counter them. In later sections, the possible solutions are treated more thoroughly.

Accuracy

Most tensor-based methods in literature use fourth-order statistics for BSI and signal separation. This choice is motivated by practical considerations. First, accurately estimating statistics is harder with increasing order [29], which implies that we prefer statistics of lower orders. Second, statistics of odd orders are zero if the probability density function of the data is even, which is often the case in practice. This leaves us with two good candidates: second- and fourth-order statistics. Because second-order statistics only yield a matrix for i.i.d. inputs, using just the second-order information does not allow us to uniquely identify a mixing system. Consequently, most methods turn to fourth-order statistics, possible combined with second-order statistics to improve the estimation accuracy. For instantaneous mixtures, several approaches to combine these statistics have been followed. We briefly review their strengths and weaknesses:

- *Methods using prewhitening* [28, 22]. These methods first use second-order statistics to prewhiten the data by principal component analysis. In a subsequent step, fourth-order statistics are used for the actual separation of the data, which then involves estimating an orthogonal mixing matrix.

This traditional prewhitening approach has the disadvantage that errors introduced in the prewhitening step, for instance by additive Gaussian noise, cannot be fully compensated for in the separation step [21, 40].

- *Methods using only fourth-order statistics* [18, 35]. This approach avoids the possible noise influence on the prewhitening step by not using second-order statistics altogether. In cases where the noise influence on the second-order statistics is significant, this will yield more accurate results.

- *Methods using soft prewhitening* [136, 163, 42]. Soft prewhitening tries to find a middle ground between the two other approaches by exploiting second- and fourth-order information simultaneously in a weighted fashion. The system can be identified by optimizing the function

$$\begin{aligned} \arg\min \quad & \omega_2 \, ||\text{Rel. error on 2nd-order statistics}||_{\mathrm{F}}^2 + \\ & \omega_4 \, ||\text{Rel. error on 4th-order statistics}||_{\mathrm{F}}^2 . \end{aligned}$$

  In this respect, traditional prewhitening uses a large ratio ω_2/ω_4 whereas fourth-order methods set $\omega_2 = 0$. As mentioned in the introduction, tensor-based methods can combine the second- and fourth-order information using coupled decompositions. More elaborate weighting strategies, including statistically optimal weighting, have been studied in for instance [128, 127, 20, 19, 98].

With these approaches for instantaneous mixtures in mind, we consider which are suitable for convolutive mixtures. Most tensor-based methods for BSI in the literature use only fourth-order statistics [59, 147]. To improve the accuracy of these methods, one has to turn to prewhitening or coupled decompositions. The prewhitening approach has been considered in [72, 9, 122], among others. In convolutive mixtures, the prewhitening step reduces the problem to the blind estimation of paraunitary channels. Approaches that couple both second- and fourth-order information in a weighted fashion in the time domain are scarce in the literature. Apart from a similar approach in the frequency domain in [24], we have not found any real results for convolutive mixtures. This approach is interesting though, since it combines the advantages of coupling statistics without resorting to the more complex statistically optimal weighting. Because of this, we will focus on methods that couple both second- and fourth-order information in a weighted fashion in the time domain. Mathematical derivations and further details will be provided in Section 4.4.

Expensive statistics computation

The spatio-temporal cumulant tensor from (4.4) contains $\mathcal{O}(M^4L^3)$ entries. For long signals or systems with many outputs, constructing this tensor is computationally expensive. For instance, computing one entry of a fourth-order cumulant tensor for signals of length N takes roughly $12N$ floating point operations. Constructing the full spatio-temporal cumulant tensor thus takes $\mathcal{O}(M^4L^3N)$ flops.

To reduce the number of computations, several strategies can be used. First, one can deconvolve the convolutive mixture to obtain an instantaneous mixture of independent components, see [141] and references therein. This is only possible when the system is strictly overdetermined, i.e., has strictly more outputs M than inputs R, but reduces the cumulant computation to $\mathcal{O}(R^4N)$ flops. The downside of this approach is that possible errors in the deconvolution step may perpetuate [141]. Second, one could compute just a subset of the statistics. This leads to an incomplete tensor, which can still be decomposed. We will elaborate on this in Section 4.5.

4.4 Coupled decompositions

To combine second- and fourth-order information, we turn to coupled tensor decompositions. For convolutive mixtures of independent components, there are several possible approaches. First, we discuss the approach that starts by deconvolving the output signals, which yields an instantaneous mixture. We can then rely on existing methods for coupling statistics. Next, an alternative is presented that immediately couples the statistics without a deconvolution step. This approach allows coupling of statistics of underdetermined systems as well.

4.4.1 Instantaneous mixture after deconvolution

One approach starts by deconvolving the convolutive mixture, which yields an instantaneous mixture of independent components [91, 138]. Note that this deconvolution step is only possible when there are strictly more system outputs than inputs. Once the instantaneous mixture is obtained, we can simply use existing techniques for coupling second- and fourth-order statistics as described in [163]. We briefly explain how this coupling is done for instantaneous mixtures for future reference.

Let $\mathbf{x}(t) \in \mathbb{C}^M$ be an instantaneous mixture of R statistically independent components stored in $\mathbf{s}(t) \in \mathbb{C}^R$ at time instance t. The mixing system can be described by a matrix $\mathbf{M} \in \mathbb{C}^{M \times R}$. We have

$$\mathbf{x}(t) = \mathbf{M}\mathbf{s}(t).$$

The covariance matrix $\mathbf{C}_{\mathbf{x}}^{(2)}$ and fourth-order cumulant tensor $\mathcal{C}_{\mathbf{x}}^{(4)}$ of the mixture $\mathbf{x}(t)$ then both admit a rank-R decomposition:

$$\mathbf{C}_{\mathbf{x}}^{(2)} = \mathbf{C}_{\mathbf{s}}^{(2)} \cdot_1 \mathbf{M} \cdot_2 \mathbf{M}^*,$$

$$\mathcal{C}_{\mathbf{x}}^{(4)} = \mathcal{C}_{\mathbf{s}}^{(4)} \cdot_1 \mathbf{M} \cdot_2 \mathbf{M}^* \cdot_3 \mathbf{M} \cdot_4 \mathbf{M}^*,$$

in which $\mathbf{C}_{\mathbf{s}}^{(2)}$ and $\mathcal{C}_{\mathbf{s}}^{(4)}$ are diagonal because of the statistical independence of the components. We can thus write

$$\mathbf{C}_{\mathbf{x}}^{(2)} = [\![\boldsymbol{\nabla}_s; \mathbf{M}, \mathbf{M}^*]\!]$$

$$\mathcal{C}_{\mathbf{x}}^{(4)} = [\![\boldsymbol{\gamma}_s; \mathbf{M}, \mathbf{M}^*, \mathbf{M}, \mathbf{M}^*]\!],$$

in which $\boldsymbol{\nabla}_s$ contains the source variances and $\boldsymbol{\gamma}_s$ the kurtosis values of the different sources. The mixture can then be identified by coupling both decompositions in one (weighted) objective function

$$\left(\widehat{\mathbf{M}}, \widehat{\boldsymbol{\nabla}}_s, \widehat{\boldsymbol{\gamma}}_s\right) = \arg\min \omega_2 \frac{\left|\left| \mathbf{C}_{\mathbf{x}}^{(2)} - [\![\widehat{\boldsymbol{\nabla}}_s; \widehat{\mathbf{M}}, \widehat{\mathbf{M}}^*]\!] \right|\right|_{\mathrm{F}}^2}{\left|\left| \mathbf{C}_{\mathbf{x}}^{(2)} \right|\right|_{\mathrm{F}}^2}$$

$$+ \omega_4 \frac{\left|\left| \mathcal{C}_{\mathbf{x}}^{(4)} - [\![\widehat{\boldsymbol{\gamma}}_s; \widehat{\mathbf{M}}, \widehat{\mathbf{M}}^*, \widehat{\mathbf{M}}, \widehat{\mathbf{M}}^*]\!] \right|\right|_{\mathrm{F}}^2}{\left|\left| \mathcal{C}_{\mathbf{x}}^{(4)} \right|\right|_{\mathrm{F}}^2}.$$

The coupling stems from the shared factor $\widehat{\mathbf{M}}$ in both decompositions. Optimizing this objective function in the variables $\widehat{\mathbf{M}}$, $\widehat{\boldsymbol{\nabla}}_s$ and $\widehat{\boldsymbol{\gamma}}_s$ yields an estimate for the mixing matrix $\widehat{\mathbf{M}}$. Various algorithms for coupled decompositions have been developed in literature [123, 5, 85, 114, 117, 153]. The choice of ω_2 and ω_4 will be discussed in Section 4.4.3.

4.4.2 Convolutive mixtures

We now wish to port the idea of combining second- and fourth-order statistics to the convolutive case. To do this, we first need tensor expressions for both that can be easily coupled. We will also briefly discuss the possible initialization procedures.

Fourth-order statistics

The fourth-order information has been derived in Section 4.3 and is contained in a fourth-order tensor admitting a structured CPD. This structured decomposition can be perfectly solved using optimization-based techniques [59, 156], though the implementation of the full structure may be cumbersome and finding a good initialization may be challenging. We first exploit the structure available in the spatio-temporal fourth-order statistics as in [147], which has proven to be a time-efficient and reasonably accurate method. Exploiting the structure by applying the algorithm from [147] results in a third-order tensor $\mathcal{Q} \in \mathbb{C}^{M(L+1)\times M(L+1)\times R}$ which can be written as

$$\mathcal{Q} = [\![\mathbf{P}^*, \mathbf{P}, \mathbf{A}]\!], \tag{4.6}$$

in which $\mathbf{P}$ is the same system matrix as in (4.5) and $\mathbf{A} \in \mathbb{C}^{R\times R}$ is a nonsingular matrix. Both $\mathbf{P}$ and $\mathbf{A}$ are found by computing the CPD of $\mathcal{Q}$, though we only need the former. The details on how to obtain this decomposition can be found in [147]. For now, it suffices to note that this tensor decomposition is unstructured and still attains a high accuracy.

Second-order statistics

To compute the second-order statistics, we first write (4.2) in matrix form as

$$\mathbf{Y} = \mathbf{P}_{\text{hor}}\mathbf{S} + \mathbf{V}, \tag{4.7}$$

in which the rows of $\mathbf{Y} \in \mathbb{C}^{M\times N}$ contain the output signals, $\mathbf{V}$ contains the additive noise, $\mathbf{P}_{\text{hor}}$ contains the system coefficients and $\mathbf{S} \in \mathbb{C}^{R(L+1)\times N}$ contains the lagged input signals. Note that $\mathbf{S}$ is block-Toeplitz structured. More specifically, its nth column $\mathbf{s}(n)$ is given by

$$\begin{aligned}\mathbf{s}(n) = &[s_1(n), s_2(n), \ldots, s_R(n),\\ &s_1(n-1), s_2(n-1), \ldots, s_R(n-1), \ldots\\ &s_1(n-L), s_2(n-L) \ldots, s_R(n-L)]^{\mathrm{T}}.\end{aligned}$$

The matrix $\mathbf{P}_{\text{hor}}$ is strongly related to $\mathbf{P}$ from (4.5) since it consists of the same submatrices, but stacked horizontally rather than vertically:

$$\mathbf{P}_{\text{hor}} = \left[\mathbf{P}^{(0)}, \mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(L)}\right] \in \mathbb{C}^{M\times R(L+1)}.$$

The covariance of the output signal $\mathbf{Y}$ can now be written as

$$\mathbf{C}_{\mathbf{y}}^{(2)} = \mathbf{P}_{\text{hor}}\mathbf{C}_{\mathbf{s}}^{(2)}\mathbf{P}_{\text{hor}}^{\mathrm{H}} + \mathbf{C}_{\mathbf{v}}^{(2)}, \tag{4.8}$$

in which $\mathbf{C}_{\mathbf{s}}^{(2)}$ and $\mathbf{C}_{\mathbf{v}}^{(2)}$ are diagonal matrices because the inputs and additive noise signals are i.i.d.

Note that (4.8) represents the standard second-order statistics rather than the spatio-temporal variant, which we used for the fourth-order statistics. The reason for this lies in simplicity of the factorization, which would be highly structured in the spatio-temporal case. However, even though we do not need the spatio-temporal second-order statistics, they can still be useful to get a noise level estimate when the system has more outputs than inputs. Consider for instance the augmented output vector $\mathbf{y}_{\text{aug}}(n) \in \mathbb{C}^{ML'}$, noise vector $\mathbf{v}_{\text{aug}}(n) \in \mathbb{C}^{ML'}$ and input vector $\mathbf{s}_{\text{aug}}(n) \in \mathbb{C}^{RL_{\text{tot}}}$ defined as

$$\begin{aligned}
\mathbf{y}_{\text{aug}}(n) = &[y_1(n), \ldots, y_M(n), y_1(n-1), \ldots, \\
&y_M(n-1), \ldots, y_M(n-L'+1)]^{\text{T}}, \\
\mathbf{s}_{\text{aug}}(n) = &[s_1(n), \ldots, s_R(n), s_1(n-1), \ldots, s_R(n-1), \\
&s_1(n-L_{\text{tot}}+1), \ldots, s_R(n-L_{\text{tot}}+1)]^{\text{T}}, \\
\mathbf{v}_{\text{aug}}(t) = &[v_1(t), \ldots, v_M(t), v_1(t-1), \ldots, \\
&v_M(t-1), \ldots, v_M(t-L'+1)]^{\text{T}},
\end{aligned}$$

In these expressions, $L_{\text{tot}} = L + L'$ and L' is an integer chosen such that $ML' > RL_{\text{tot}}$. The augmented vectors are related by

$$\mathbf{y}_{\text{aug}}(n) = \mathbf{H}\mathbf{s}_{\text{aug}}(n) + \mathbf{v}_{\text{aug}}(t),$$

with $\mathbf{H} \in \mathbb{C}^{ML' \times RL_{\text{tot}}}$ a tall structured matrix containing the system coefficients. The exact structure of $\mathbf{H}$ can be easily derived from (4.7), as was done in [13], but is not important for our purposes. Computing the covariance matrix of $\mathbf{y}_{\text{aug}}(n)$, which is equivalent to the spatio-temporal covariance of $\mathbf{y}(n)$, yields

$$\mathbf{C}_{\mathbf{y}_{\text{aug}}}^{(2)} = \mathbf{H}\mathbf{C}_{\mathbf{s}_{\text{aug}}}^{(2)}\mathbf{H}^{\text{H}} + \mathbf{C}_{\mathbf{v}_{\text{aug}}}^{(2)}. \tag{4.9}$$

Just as before, the matrices $\mathbf{C}_{\mathbf{s}_{\text{aug}}}^{(2)}$ and $\mathbf{C}_{\mathbf{v}_{\text{aug}}}^{(2)}$ are diagonal since both inputs and additive noise signals are i.i.d. A key observation here is that the rank of $\mathbf{H}\mathbf{C}_{\mathbf{s}_{\text{aug}}}^{(2)}\mathbf{H}^{\text{H}}$ equals RL_{tot}, which is strictly smaller than ML'. Assuming the additive noise levels are the same for all output signals, this implies that the $ML' - RL_{\text{tot}}$ smallest singular values of $\mathbf{C}_{\mathbf{y}_{\text{aug}}}^{(2)}$ are solely determined by the noise variance σ_v^2. We can thus estimate the noise level by computing a singular value decomposition of $\mathbf{C}_{\mathbf{y}_{\text{aug}}}^{(2)}$ and considering the smallest $ML' - RL_{\text{tot}}$ singular values. This noise variance can then be used improve the estimate of

the output covariance in (4.8) by subtracting a diagonal matrix containing the noise variances $\mathbf{C}_{\mathbf{v}}^{(2)} = \sigma_v^2 \mathbf{I}$:

$$\widetilde{\mathbf{C}}_{\mathbf{y}}^{(2)} = \mathbf{C}_{\mathbf{y}}^{(2)} - \mathbf{C}_{\mathbf{v}}^{(2)} = \mathbf{P}_{\text{hor}} \mathbf{C}_{\mathbf{s}}^{(2)} \mathbf{P}_{\text{hor}}^{\text{H}}. \tag{4.10}$$

As a sidenote, we mention that equation (4.9) was used for prewhitening in [13], in the context of the blind identification of non-i.i.d. system inputs.

Coupled decomposition

We can now combine second- and fourth-order information in the following coupled decomposition:

$$\begin{aligned} \widehat{\mathbf{P}} = \arg\min \quad & \omega_2 \frac{\left|\left| \widetilde{\mathbf{C}}_{\mathbf{y}}^{(2)} - \widehat{\mathbf{P}}_{\text{hor}} \widehat{\mathbf{P}}_{\text{hor}}^{\text{H}} \right|\right|_{\text{F}}^2}{\left|\left| \widetilde{\mathbf{C}}_{\mathbf{y}}^{(2)} \right|\right|_{\text{F}}^2} \\ & + \omega_4 \frac{\left|\left| \mathcal{Q} - \left[\!\left[\widehat{\mathbf{P}}^*, \widehat{\mathbf{P}}, \widehat{\mathbf{A}} \right]\!\right] \right|\right|_{\text{F}}^2}{||\mathcal{Q}||_{\text{F}}^2}. \end{aligned} \tag{4.11}$$

Note that we assumed that the diagonal matrix $\widetilde{\mathbf{C}}_{\mathbf{s}}^{(2)}$ from (4.8) has all ones on the diagonal. This can be done without loss of generality since the column scaling effects introduced by this can be captured in the factor $\mathbf{A}$. If the noise variance cannot be estimated properly, which is the case if the system is not strictly overdetermined, we use $\mathbf{C}_{\mathbf{y}}^{(2)}$ instead of $\widetilde{\mathbf{C}}_{\mathbf{y}}^{(2)}$. Though the second-order statistics are less accurate in this case, it may still be beneficial to combine both second- and fourth-order statistics since the former could still add information. The coupled optimization problem can be initialized using the algebraic method from [147], which uses only fourth-order statistics to find a solution. It has been shown in [147] that this approach is both fast and reasonably accurate. We illustrate this in Section 4.6 as well.

4.4.3 Setting the weights

The weights ω_2 and ω_4 in (4.11) have to be chosen by the user. Since the fourth-order information ensures uniqueness of the coupled decomposition, the only limitation is that ω_4 must be larger than zero. Choosing a large ratio

Algorithm 3: High-level overview of the procedure to combine fourth- and second-order information for blind system identification.

Data: M output data sequences $y_m(n)$.
Result: Estimate of the mixing channels $\widehat{\mathbf{P}}$.

1. Compute the full fourth-order spatio-temporal cumulant tensor $\mathcal{T}$ as in (4.3) and (4.4).
2. Exploit the block-Hankel structure of the factor matrices of $\mathcal{T}$ to obtain the smaller tensor $\mathcal{Q}$ in (4.6) as explained in [147].
3. Compute the CPD of $\mathcal{Q}$ to obtain an initial estimate $\widehat{\mathbf{P}}_{\text{init}}$ from (4.6).
4. Compute the covariance of the output signal in (4.8).
5. If the system is overdetermined, compute the noise variance $\mathbf{C}_{\mathbf{v}}^{(2)}$. Compensate for the noise in the covariance matrix as shown in (4.10).
6. Couple the fourth-order information $\mathcal{Q}$ and the second-order information as in (4.11). The optimization can be initialized using $\widehat{\mathbf{P}}_{\text{init}}$.

ω_4/ω_2 indicates that one has more confidence in the fourth-order statistics. This can be the case when many samples are available and the additive noise is Gaussian. A small ratio ω_4/ω_2 implies a strong reliance on the second-order statistics, which is useful when few samples are available or when the noise influence on the second-order statistics was corrected.

There is an optimal choice of weights which can be derived from Fisher information theory. However, because this optimum depends on the a priori unknown mixing coefficients, source statistics, estimator statistics and noise statistics, the optimal value is hard to determine in practice. A full mathematical derivation is out of the scope of this paper, but we illustrate the effect of different weights through numerical experiments in Section 4.6.

4.5 Incomplete tensors

In Section 4.3.2, we suggested that incomplete tensors may be a solution to the expensive statistics computation. The underlying idea is that only a subset of the fourth-order cumulant entries is computed, yielding an incomplete tensor. One interesting property is that these incomplete tensors can still be decomposed

if there are enough known entries. Algorithms for this are available in the literature [114, 129, 3, 76, 155, 152, 120].

Choosing the number of entries to compute is a trade-off between computation time and accuracy of the solution. This balance is more intricate than one would expect at first sight, since the computational load is not only influenced by the number of statistics that have to be computed, but is affected by the initialization procedure and algorithm as well. This trade-off between computation time and solution accuracy will be illustrated by a numerical experiment in Section 4.6.

Since the available algorithms for decomposing incomplete tensors are optimization-based, a good initialization is key to avoid getting stuck in local minima. In the case of full tensors, algebraic techniques can be used to obtain a proper initial value. For instance, an algebraic method for BSI using a tensor of fourth-order statistics is presented in [147]. One of the main issues with incomplete tensors is that this straightforward algebraic technique can no longer be used, though we mention for completeness that sampling techniques exist that do allow algebraic computations [120]. We propose a two-step procedure that uses an algebraic initialization and subsequently refines the result using an incomplete tensor, capitalizing on the fact that fourth-order statistics are computationally expensive for long signals. Suppose we have M system outputs of length N. Instead of computing the full spatio-temporal cumulant tensor as in Section 4.3, we use just a part of the available output data. More specifically, we do not consider the full-length output signals, but select just the first fragment of each output signal. The goal is to get a reasonable first estimate for the spatio-temporal cumulant tensor without using the full signals. Depending on the original length N of the signals, the first 1% or even 0.1% of the signal samples may suffice to get a reasonable estimate. If more samples are taken into account at this stage, the estimate is expected to be more accurate at a higher computational cost. Note that it does not matter from which part of the output data a subset is selected because the input signals are i.i.d. Once we have the full spatio-temporal cumulant tensor based on these signal segments, existing algebraic methods can be used to obtain a first estimate of the system parameters [147]. In a next step, we construct a subset of the expensive cumulant tensor entries using the full-length signals, yielding an incomplete tensor. Optimization-based algorithms can use this incomplete tensor to refine the initialization obtained in the first step.

The overall objective of this two-step strategy is to obtain a solution in a much faster way than using the full spatio-temporal cumulant tensor, while only having a limited loss in accuracy. Note that both steps in this strategy rely on partial information, but in a different way as illustrated graphically in Figure 4.2. To summarize, a high-level overview of the approach is given in Algorithm-4. For completeness we mention that this two-step principle can be applied for

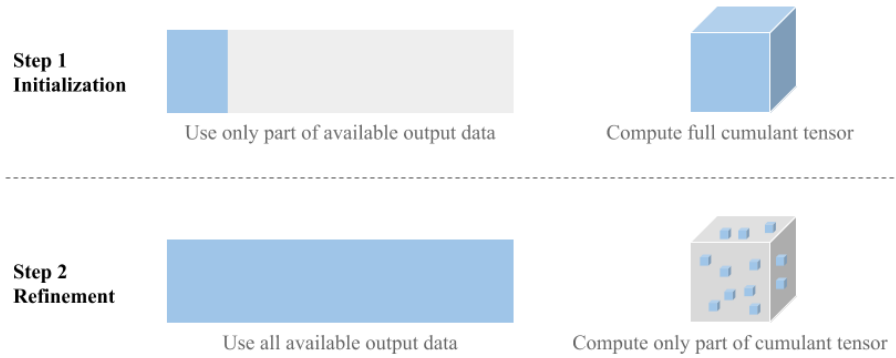


Figure 4.2: Graphical illustration of the two-step strategy using incomplete tensors.

input signals that are not i.i.d. as well, though a different cumulant tensor and decomposition algorithm is needed in that case.

Algorithm 4: High-level overview of the procedure using incomplete tensors for blind system identification.

Data: M output data sequences $y_m(n)$.
Result: Estimate of the mixing channels $\widehat{\mathbf{P}}$.

1. Construct the full fourth-order spatio-temporal cumulant tensor $\mathcal{T}$ as in (4.3) and (4.4) using only part of the available observations of $y_m(n)$.
2. Exploit the block-Hankel structure of the factor matrices of $\mathcal{T}$ to obtain the smaller tensor $\mathcal{Q}$ in (4.6) as explained in [147].
3. Compute the CPD of $\mathcal{Q}$ to obtain an initial estimate $\widehat{\mathbf{P}}_{\text{init}}$ from (4.6).
4. Construct a subset of the entries of the fourth-order cumulant tensor $\mathcal{T}$ as in (4.4) using all available observations of $y_m(n)$. Store these values in $\mathcal{T}_{\text{incomplete}}$.
5. Compute the structured CPD of $\mathcal{T}_{\text{incomplete}}$ using the initialization $\widehat{\mathbf{P}}_{\text{init}}$.

In constructing the incomplete tensor, one has to choose which cumulant entries to compute and which to consider missing. This choice may have an influence on the accuracy of the result. A full derivation is out of the scope of this paper, but we will perform a numerical experiment illustrating some effects in Section 4.6.

4.6 Numerical experiments

In this section, the performance of the presented methods is shown and compared to other approaches. First, we show how the choice of weights in a coupled decomposition affects the accuracy. Next, we compare coupled decompositions with pure fourth-order methods. Finally, we illustrate the effectiveness of incomplete cumulant tensors for BSI.

To measure the estimation accuracy, we use the relative error of the system coefficients, which we define as

$$\text{Relative error}(\mathbf{P}) = 20\log_{10}\left(\frac{\left|\left|\mathbf{P} - \widehat{\mathbf{P}}\boldsymbol{\Delta}_{\text{opt}}\boldsymbol{\Pi}_{\text{opt}}\right|\right|_{\text{F}}}{||\mathbf{P}||_{\text{F}}}\right) \text{(dB)},$$

in which $\boldsymbol{\Delta}_{\text{opt}}$ and $\boldsymbol{\Pi}_{\text{opt}}$ represent optimal column scaling and permutation, which can be determined using the `cpderr` command in Tensorlab [156]. All other tensor-related decompositions and operations are implemented using Tensorlab as well.

4.6.1 Choosing the weights for coupling

In Section 4.4.3 we discussed the choice of the relative weights ω_2 and ω_4 for the second-order and fourth-order statistics, respectively. In Figure 4.3, we illustrate the effect of the choice of weights for two systems. Both of these FIR systems have a maximum filter delay $L = 1$ and their outputs are perturbed by additive Gaussian noise such that the signal-to-noise ratio (SNR) equals 20 dB. The input signals are sampled randomly from a uniform distribution on $[-\frac{1}{2}, \frac{1}{2}]$ and the system coefficients are sampled randomly from a standard normal distribution. The weight ω_2 is varied and ω_4 is computed as $\omega_4 = 1 - \omega_2$. The figure shows the median accuracy over 1000 experiments.

The top graph of Figure 4.3 shows a system with 4 outputs and 4 inputs with signals consisting of 10^5 samples. Since there are many samples available and the additive noise is Gaussian, we expect that the fourth-order statistics will be approximated well. On another note, the system is not strictly overdetermined, which implies that we cannot subtract the noise variance from the covariance matrix. Combining these insights and using our knowledge from Section 4.4.3, we expect that a small weight for the second-order statistics will be better for this system. Indeed, the top of Figure 4.3 shows that the best result is obtained when we only use fourth-order statistics. The accuracy gradually deteriorates when increasing the weight of the second-order statistics. For $\omega_2 = 1$, the

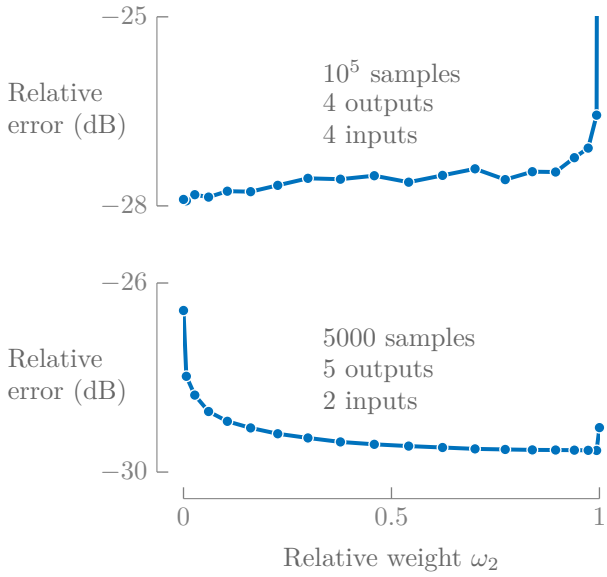


Figure 4.3: The effect of different a relative weight ω_2 depends on the system and signal parameters. The graphs show the median over 1000 experiments at an SNR of 20 dB of a system with $L = 1$.

system estimation fails completely since the covariance matrix alone does not suffice to uniquely identify a system.

The bottom of Figure 4.3 shows a system with 5 outputs and 2 inputs with signals consisting of 5000 samples. Due to the limited number of samples, the fourth-order statistics will not be estimated very accurately. However, the noise influence on the covariance matrix can be estimated since the system is strictly overdetermined. This implies that the second-order information will be estimated quite accurately. The figure confirms that a higher weight for the second-order statistics improves the accuracy of the system identification. Again, this holds up to $\omega_2 = 1$, which does no longer allow to uniquely identify the system.

Both experiments were conducted at a fixed SNR of 20 dB. However, we can make a similar reasoning as before to determine what happens if the SNR changes. For instance, a lower SNR will have little influence on the fourth-order statistics, which are theoretically blind to additive Gaussian noise, though there will be a small impact because we are dealing with finite signals. By contrast, the second-order statistics may be strongly affected if the noise variance cannot

be estimated (i.e., if the system is not strictly overdetermined) since the noise directly perturbs the diagonal of the output covariance matrix. Because of this, the fourth-order statistics would get a higher weight in the optimal coupled decomposition. For the sample size, an analogous reasoning can be used. Second-order statistics can be estimated more accurately than fourth-order statistics for a fixed number of samples [29]. Consequently, the second-order statistics will be favored more by the optimal weights if the sample size is very small.

4.6.2 Comparison of methods

In this section, we show how the methods that combine second- and fourth-order information perform when compared to other methods. For the numerical experiment, we consider a system with 5 outputs, 2 inputs and a filter length of 2, which implies that the maximum filter delay is $L = 1$. The input and output signals consist of 5000 observations. The inputs are randomly sampled from a uniform distribution on $[-\frac{1}{2}, \frac{1}{2}]$. Additive noise is added to the outputs to obtain various SNRs.

In Figure 4.4, four methods are compared in terms of accuracy and computation time. All results are the median over 300 experiments. The first method is a pure fourth-order method from [147], which immediately tensorizes the data and subsequently identifies the system by computing a structured tensor decomposition. The second method couples the method from [147] with second-order information, as explained in 4.4.2. The relative weights were chosen as $\omega_2 = 0.8$ and $\omega_4 = 0.2$. This choice is based on the results of the previous experiment, which showed that for a similar system, a large weight for the second-order information is appropriate. The third and fourth method start by deconvolving the convolutive mixture to obtain an instantaneous mixture. For the deconvolution, we used the fast approach from [141] that uses only 100 data samples for the deconvolution. It has been shown in [141] that this approach greatly reduces the computation time and has little impact on the accuracy of the deconvolution. The third method then identifies the remaining instantaneous mixture using only the fourth-order cumulant. The fourth method again couples second- and fourth-order information of this instantaneous mixture as explained in Section 4.4.1. Just as for the other algorithm, the relative weights were chosen to be $\omega_2 = 0.8$ and $\omega_4 = 0.2$.

The top of Figure 4.4 shows that the accuracy of all methods improves when the SNR increases up to a lower bound, as expected. For low SNR, the methods that immediately tensorize perform better than the methods that start with deconvolution. This is because the fourth-order statistics of the additive Gaussian noise is zero, which implies that immediately computing the (fourth-

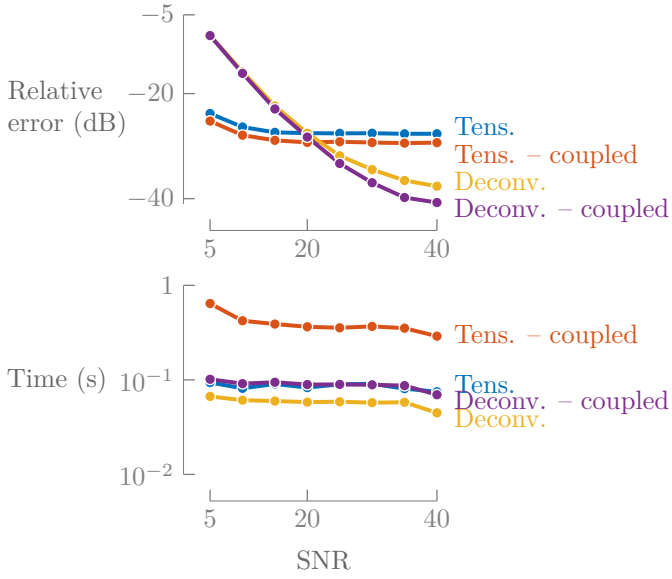


Figure 4.4: Coupling second- and fourth-order information leads to better accuracies for methods starting with both deconvolution and tensorization. This improved accuracy comes at the price of a slower method.

order) statistics has a denoising effect. Methods that start by deconvolving the outputs do not have this advantage and tend to introduce errors that cannot be fully undone in the second step of the algorithm. When the SNR is sufficiently high, the estimation accuracy of the fourth-order statistics is the limiting factor. In this case, it is better to start with deconvolving the data. The next goal of this experiment is to show that coupling second- and fourth-order information improves the results. The top of Figure 4.4 clearly shows that this is the case. Note that the coupled method that immediately tensorizes the data improves upon the pure fourth-order method by about the same factor for all SNRs, whereas the coupled method after deconvolution keeps improving for higher SNRs when compared to its fourth-order counterpart. The reason for this lies in the noise variance estimation. For the method that starts by tensorizing, the noise influence on the covariance matrix was removed. For the methods that start with deconvolving, this is not possible since the resulting instantaneous mixture is not strictly overdetermined.

The same qualitative behavior can be seen if the filter length is increased to 4, which implies that $L = 3$, as shown at the top of Figure 4.5. For this experiment with longer filters, all parameters have been kept the same as before, apart from

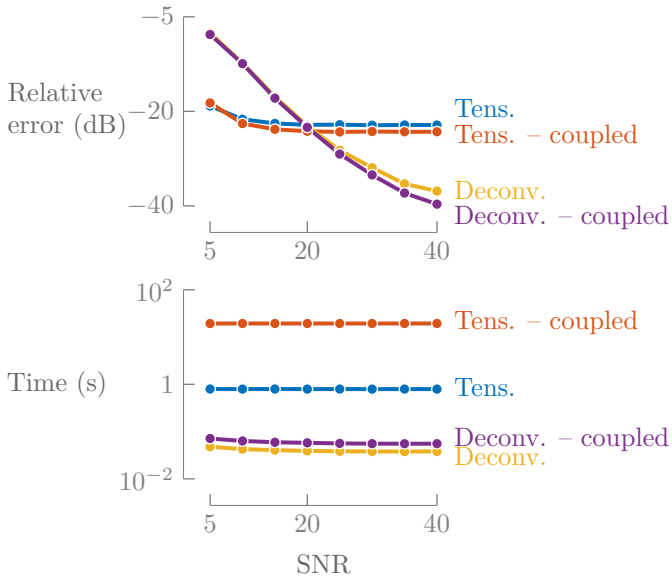


Figure 4.5: Accuracy and time complexity comparison of methods using either only fourth-order information and methods combining fourth- and second-order information, this time for slightly longer filters ($L = 3$).

the number of outputs, which has increased to $M = 7$.

The bottom of Figure 4.4 shows the time complexity of the methods implemented in Matlab version R2015b on a laptop containing an Intel Core i7-4810MQ CPU and 16 GB of RAM. It shows that the downside of coupling is the higher computational complexity. For the methods starting with deconvolution, the effect is limited since efficient algorithms exist for shared factors in coupled matrix-tensor factorizations [123, 114]. In the coupled method that starts by tensorizing the data, the factors consists of the same building blocks, but they are ordered differently as explained in Section 4.4.2. This implies that we have to turn to more general optimization frameworks or write dedicated optimization routines tailored to this problem. In this experiment, the structured data fusion framework of Tensorlab was used with `sdf_nls`, which allows very flexible modeling but is slower than dedicated algorithms [114]. For larger filters and bigger output vectors, the time complexity increases significantly for the methods that start by tensorizing as illustrated for $L = 3$ at the bottom of Figure 4.5. For high values of L, methods that deconvolve first are more suited in terms of time complexity.

Table 4.1: Classification of cumulant entries based on the repeated indices of the spatial parameters.

Entry class	Selection procedure 2 (equal preference)	Selection procedure 3 (unequal preference)
c_{iiii}	1	5
c_{iiij}	2	4
c_{iijj}	3	3
c_{iijk}	4	2
c_{ijkl}	5	1

4.6.3 Incompleteness

In this section, we compare the method based on incomplete tensors with the full method using all statistics.

Consider a system having eight outputs, maximum filter delay of $L = 2$ and 3 inputs. The signals have length $N = 10^6$ samples and Gaussian noise is added to the output signals such that the SNR is 20 dB. For the incomplete approach, an initialization is computed by decomposing the full cumulant tensor constructed using the first 10^4 samples of the output signals. The experiment is run 100 times and the median results are shown in Figure 4.6. In the figure, three selection procedures are shown. These procedures decide which entries of the incomplete cumulant tensor are computed. The first one randomly selects entries to compute. The second and third selection procedures are slightly more involved. Recall the fourth-order spatio-temporal cumulant expression (4.3). If we ignore the time-shifts, the combinations of the four spatial parameters m_1, m_2, m_3 and m_4 can be classified according to repeated indices as shown in Table 4.1. For instance, the notation c_{iiii} is used to denote cumulant entries of which the spatial parameters are all the same. Similarly, c_{iiij} denotes entries of which three spatial parameters are the same and the remaining one is different. The table then shows in which order the second and third selection procedure choose the entries to compute. They each start by randomly selecting entries from the class labeled "1" until no entries remain and then move on to the other classes until the desired number of entries is reached. Note that there are many different options to choose entries and these procedures only show a few extremes to give a general idea of the effects of different choices.

The top graph in Figure 4.6 shows that the random entry selection outperforms the other approaches in this example. This may be because the random approach allows entries from all classes given in Table 4.1. However, further research is needed to thoroughly explain this behavior. Overall, all approaches show that

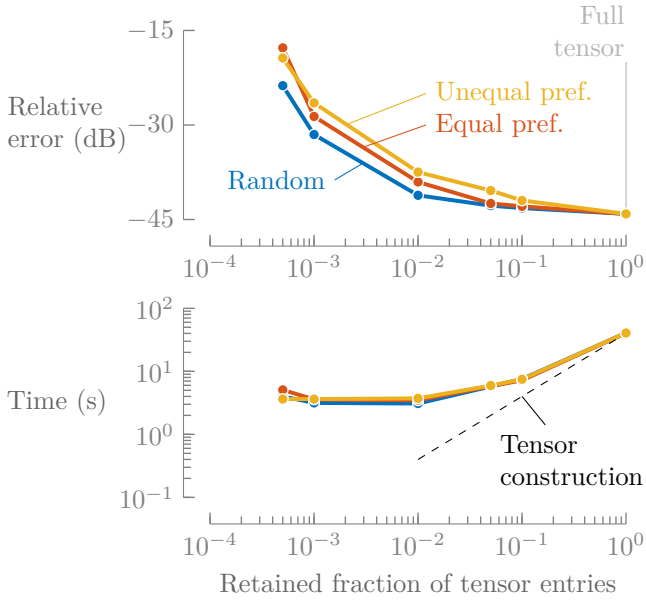


Figure 4.6: Using incomplete statistics leads to faster methods at the cost of a lower accuracy. The selection procedure of which entries to compute also impacts accuracy. The dashed black line denotes the time needed to construct the (incomplete) cumulant tensor.

using incomplete statistics leads to less accurate results than the full method, which was expected since less information is available. The advantage of the incomplete methods lies in the computation time, as shown in the bottom part of Figure 4.6. Note that in this figure, the dashed black line denotes the cumulant tensor construction time, which clearly takes up most of the time for the full method. By circumventing this full construction in our incomplete approach, results can be obtained much more quickly. In this experiment, a reasonable trade-off between accuracy and speed lies around a retained fraction of 10^{-2}. At this level of incompleteness, there is just a small loss in accuracy while there is a speedup of an order of magnitude.

4.7 Conclusion

In this paper, we described the current state of tensor-based methods for BSI with statistically independent inputs and what the drawbacks are in terms of

accuracy and time complexity. We subsequently proposed strategies to deal with these disadvantages. More specifically, coupled decompositions allow the user to combine both second- and higher-order information in convolutive mixtures of independent components, which leads to more accurate results. To deal with the time complexity, incomplete tensors can be used. By computing just a subset of the statistics needed, the execution time can be strongly reduced with only limited loss of accuracy. The results have been illustrated in various numerical experiments. Apart from the main results, experiments were performed to illustrate the effect of some parameters such as the relative weights in coupled decomposition and the choice of entries to compute in incomplete tensors.

Chapter 5

Algorithms for canonical polyadic decomposition with block-circulant factors

Abstract — Higher-order tensors and their decompositions are well-known tools in signal processing. More specifically, tensors admitting decompositions with structured factor matrices arise in various applications, such as telecommunications or convolutive independent component analysis. These applications motivate the development of efficient algorithms for structured tensor decompositions. In this paper, we develop a method for canonical polyadic decompositions (CPD) with block-circulant factor matrices by extending a method for circulant factors. Block-circulant factor matrices arise in the identification of homogeneous Wiener–Hammerstein models. By transforming the data tensor to the frequency domain, the available structure is exploited using simple element-wise divisions. This approach reformulates the structured CPD as an unstructured CPD of lower rank. The resulting CPD of a possibly incomplete tensor can then be solved algebraically or using optimization-based methods.

Reference — This chapter is a slightly adapted version of [142]. The candidate did the research and wrote the article under the guidance of the coauthor.

5.1 Introduction

Tensor decompositions are well-established tools in signal processing and are used for signal separation, data compression, completion problems and more [25, 81, 155]. The efficient computation of tensor decompositions has been the object of research for decades, and many algorithms have been developed [25, 81]. Broadly speaking, these algorithms can be divided into two classes: optimization-based algorithms and algebraic algorithms. As the name suggests, the former optimize an objective function using traditional techniques to find the decomposition [113, 104, 130, 4]. This approach can handle large tensors (e.g. [150, 101, 107, 75]) and is robust to noise, provided the algorithm is properly initialized. Algebraic algorithms decompose tensors using only algebraic operations, such as (generalized) eigenvalue decompositions or singular value decompositions, and are therefore fast for reasonably-sized tensors. In the noiseless case, algebraic algorithms are guaranteed to find the exact solution. For noisy data, they can provide a good initialization for optimization-based methods.

In various applications additional structure is present in the factors of the tensor decomposition. For instance, Vandermonde structured factors arise when using uniform linear arrays [116] and block-Hankel structured factors can be found in convolutive independent component analysis [59].

For better results, this a priori known structure can be taken into account when computing the decomposition. Because optimization-based methods may converge to a local minimum, the initialization of these dedicated algorithms is important. Since algebraic algorithms can be used to provide a proper initialization, methods for various structured tensor decompositions are receiving increasing attention [63, 78, 116, 124, 147, 115]. In this paper, we focus on tensors admitting a canonical polyadic decomposition with (block-)circulant factors. Such type of factors arise in the identification of homogeneous Wiener–Hammerstein systems [77, 63]. Circulant matrices are also used to approximate Toeplitz matrices, e.g. in frequency domain convolutive independent component analysis [103].

An algorithm for CPDs with circulant factors has already appeared in [63]. The algorithm rewrites the constrained CPD as a set of monomial equations, which can be solved to obtain the factors. The method presented in this paper stays in the tensor domain, reducing the circulant constrained CPD to an unstructured CPD with rank equal to the width of the circulant blocks. Both approaches are equivalent in terms of underlying mathematics since any rank-1 CPD can be written as a set of monomial equations. However, staying in the tensor domain allows us to easily generalize the algorithm to block-circulant structured

factor matrices. An additional advantage is that existing tensor decomposition algorithms and developed uniqueness properties can be used.

Notations

Scalars are denoted by lowercase letters (e.g., a), vectors by bold lowercase letters (e.g., $\mathbf{a}$), matrices by bold uppercase letters (e.g., $\mathbf{A}$), and tensors by uppercase calligraphic letters (e.g., $\mathcal{A}$). The outer product is denoted by $\circ$. We use $\star$ and $\oslash$ for the element-wise product and division, respectively. The discrete Fourier transform and inverse transform are respectively given by $\mathcal{F}\{\cdot\}$ and $\mathcal{F}^{-1}\{\cdot\}$. The Frobenius norm is given by $||\cdot||_{\mathrm{F}}$. Estimates are denoted by a hat atop, e.g. $\widehat{a}$.

5.2 Tensors and Problem statement

5.2.1 Canonical polyadic decomposition

An Nth-order tensor has rank 1 if and only if it equals the outer product of N nonzero vectors. By extension, the rank of a tensor $\mathcal{A}$ is defined as the minimal number of rank-1 tensors yielding $\mathcal{A}$ in a linear combination. The canonical polyadic decomposition (CPD) writes a tensor $\mathcal{A}$ as a minimal sum of rank-1 terms:

$$\mathcal{A} = \sum_{q=1}^{Q} \mathbf{u}_q^{(1)} \circ \cdots \circ \mathbf{u}_q^{(N)} = \left[\!\left[\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\right],$$

in which factor matrix $\mathbf{U}^{(n)}$ has Q columns $\mathbf{u}_q^{(n)}$ [81, 25].

Contrarily to a decomposition of a matrix in rank-1 terms, the CPD of higher-order tensors is essentially unique under fairly mild conditions [84, 108, 50]. Essential uniqueness means that the decomposition is unique up to trivial indeterminacies. More specifically, the different rank-1 terms can be arbitrarily permuted and the vectors within each rank-1 term may be scaled and counterscaled as long as the rank-1 term remains unchanged.

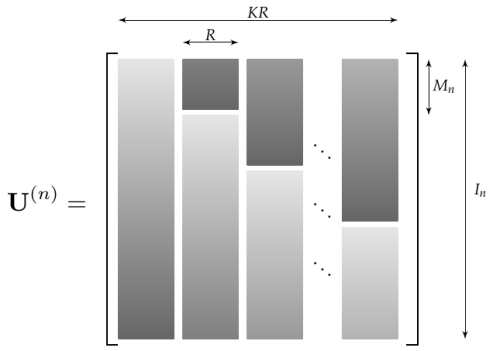


Figure 5.1: Block-circulant structure of the factor matrix $\mathbf{U}^{(n)}$.

5.2.2 Problem statement

Consider an Nth-order tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ admitting a CPD of rank KR:

$$\mathcal{T} = \sum_{r=1}^{KR} \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(N)} = \left[\!\!\left[\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(N)} \right]\!\!\right].$$

Assume all factors $\mathbf{U}^{(n)} \in \mathbb{C}^{I_n \times KR}$ have block-circulant structure with blocks having R columns and circular shifts of size M_n, as depicted in Figure 5.1. Note that, apart from the block-circulant structure, the columns may be arbitrary. Let $\mathbf{U}_k^{(n)} \in \mathbb{C}^{I_n \times R}$ denote a block-column of $\mathbf{U}^{(n)}$ such that $\mathbf{U}^{(n)} = [\mathbf{U}_1^{(n)}, \mathbf{U}_2^{(n)}, \ldots, \mathbf{U}_K^{(n)}]$. If $\mathbf{U}_1^{(n)}$ is known, the matrix $\mathbf{U}^{(n)}$ is fully determined because of the block-circulant structure. The aim of the presented method is to efficiently compute the generating blocks $\mathbf{U}_1^{(n)}$, for $n \in \{1, \ldots, N\}$.

5.3 Algorithm and uniqueness

As in [63], we start by computing the multidimensional discrete Fourier transformation (DFT) of $\mathcal{T}$, which we denote by $\mathcal{T}^{(\mathcal{F})}$. Since this transformation is multilinear, the multidimensional DFT can be interpreted as the column-wise DFT of the factor matrix in each mode. Using the shift property of the DFT and the block-circulant structure of $\mathbf{U}^{(n)}$, the DFT of the rth column of the kth block-column of $\mathbf{U}^{(n)}$ can be written as

$$\mathcal{F}\left\{ \left(\mathbf{U}_k^{(n)} \right)_r \right\} = \mathbf{g}_r^{(n)} \star \boldsymbol{\alpha}^{M_n(k-1)/I_n}, \tag{5.1}$$

in which

$$\boldsymbol{\alpha}^{M_n(k-1)/I_n} = \begin{bmatrix} 1 \\ e^{-2\pi j M_n(k-1)/I_n} \\ e^{-4\pi j M_n(k-1)/I_n} \\ \vdots \\ e^{-2(I_n-1)\pi j M_n(k-1)/I_n} \end{bmatrix} \in \mathbb{C}^{I_n},$$

and $\mathbf{g}_r^{(n)}$ is the rth column of $\mathbf{G}^{(n)}$, given by

$$\mathbf{G}^{(n)} = \mathcal{F}\left\{\mathbf{U}_1^{(n)}\right\} \in \mathbb{C}^{I_n \times R}.$$

Using expression (5.1), the CPD of $\mathcal{T}^{(\mathcal{F})}$ can be written as

$$\mathcal{T}^{(\mathcal{F})} = \sum_{r=1}^{R}\sum_{k=1}^{K}\left(\mathbf{g}_r^{(1)} \star \boldsymbol{\alpha}^{M_1(k-1)/I_1}\right) \circ \cdots \circ \left(\mathbf{g}_r^{(N)} \star \boldsymbol{\alpha}^{M_N(k-1)/I_N}\right).$$

A single entry of $\mathcal{T}^{(\mathcal{F})}$ is given by

$$\begin{aligned} t_{i_1 i_2 \cdots i_N}^{(\mathcal{F})} &= \sum_{r=1}^{R}\sum_{k=1}^{K} g_{i_1 r}^{(1)} g_{i_2 r}^{(2)} \cdots \\ &\quad g_{i_N r}^{(N)} e^{-2\pi j i_1 M_1(k-1)/I_1} e^{-2\pi j i_2 M_2(k-1)/I_2} \cdots e^{-2\pi j i_N M_N(k-1)/I_N} \\ &= \left(\sum_{r=1}^{R} g_{i_1 r}^{(1)} g_{i_2 r}^{(2)} \cdots g_{i_N r}^{(N)}\right) \cdot \\ &\quad \left(\sum_{k=1}^{K} e^{-2\pi j (M_1(k-1)i_1/I_1 + M_2(k-1)i_2/I_2 + \cdots + M_N(k-1)i_N/I_N)}\right) \end{aligned} \tag{5.2}$$

Note that the summation over k is fully determined by the known structure of the factor matrices. To simplify notation, rewrite (5.2) as

$$t_{i_1 i_2 \cdots i_N}^{(\mathcal{F})} = a^{(i_1, i_2, \cdots, i_N)} \sum_{r=1}^{R} g_{i_1 r}^{(1)} g_{i_2 r}^{(2)} \cdots g_{i_N r}^{(N)}, \tag{5.3}$$

in which $a^{(i_1, i_2, \cdots, i_N)}$ represents the factor containing the summation over k. Because this factor is known in advance, it can be constructed separately for all tensor entries. To do so, construct the matrices $\mathbf{A}^{(n)} \in \mathbb{C}^{I_n \times K}$ given by

$$\mathbf{A}^{(n)} = \begin{bmatrix} \mathbf{1} & \boldsymbol{\alpha}^{M_n/I_n} & \cdots & \boldsymbol{\alpha}^{M_n(K-1)/I_n} \end{bmatrix}.$$

The tensor $\mathcal{A} = \left[\!\left[\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)} \right]\!\right] \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ then has entries

$$a_{i_1 i_2 \cdots i_N} = a^{(i_1, i_2, \cdots, i_N)}. \tag{5.4}$$

It follows from (5.3) and (5.4) that $\mathcal{T}^{(\mathcal{F})}$ can be written as

$$\mathcal{T}^{(\mathcal{F})} = \mathcal{G} \star \mathcal{A},$$

in which $\mathcal{G} = \left[\!\left[\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \cdots, \mathbf{G}^{(N)} \right]\!\right]$ is a rank-R tensor. The rest of the algorithm depends on whether any entries of $\mathcal{A}$ are zero. If all entries are nonzero, we can simply divide $\mathcal{T}^{(\mathcal{F})}$ element-wise by $\mathcal{A}$ to obtain $\mathcal{G}$:

$$\mathcal{G} = \mathcal{T}^{(\mathcal{F})} \oslash \mathcal{A}. \tag{5.5}$$

By doing so, we have exploited the circulant structure of the factors and have reduced the rank from RK to R. The factors of $\mathcal{G}$ can then be obtained using any algorithm that is able to handle complex data. If the data is noiseless, computing this CPD using a generalized eigenvalue decomposition (GEVD) yields a fully algebraic method [51]. In noisy conditions, iterative (refinement) algorithms typically give the best results. Note that the uniqueness conditions for this unstructured decomposition of $\mathcal{G}$ are equivalent to the uniqueness conditions for the structured decomposition of $\mathcal{T}$, which allows us to verify these conditions easily [50]. Once we have an estimate for the factors $\widehat{\mathbf{G}}^{(n)}$, we can retrieve the generators of the original factors up to scaling and permutation by computing the inverse column-wise DFT

$$\widehat{\mathbf{U}}_1^{(n)} = \mathcal{F}^{-1} \left\{ \widehat{\mathbf{G}}^{(n)} \right\}. \tag{5.6}$$

If the full factor matrices are needed, they can easily be constructed using the generators $\widehat{\mathbf{U}}_1^{(n)}$.

Note that this approach is only possible if none of the entries of $\mathcal{A}$ is zero. In [63], a proposition is derived establishing a necessary and sufficient condition for nonnullity of the entries of $\mathcal{A}$ for circulant factors. For our purposes, it suffices to check the tensor $\mathcal{A}$ for zero values. If $a^{(i_1, i_2, \cdots, i_N)}$ is zero, we will treat the corresponding entry of $\mathcal{G}$ as missing. In this way, we obtain an incomplete tensor, of which the CPD can still be computed provided enough entries are available. Decompositions of incomplete tensors can be computed using optimization-based methods [25, 155, 17]. If there are at least two slices of $\mathcal{G}$ that do not contain any missing values, a GEVD can be computed of the subtensor consisting of these slices to partially initialize the optimization-based method. From the decomposition of $\mathcal{G}$ an estimate $\widehat{\mathbf{G}}^{(n)}$ is obtained, which can be used to find the original circulant factors as described above in (5.6).

To recap, our main method computes the CPD of $\mathcal{G}$ using a GEVD, unless $\mathcal{G}$ is incomplete, in which case an optimization-based method is used. Note that instead of explicitly dividing in (5.5) and computing a CPD of a tensor with missing entries, a weighted CP decomposition of $\mathcal{T}^{(\mathcal{F})}$ can be used instead.

The time complexity of the presented method (with a generalized eigenvalue decomposition for the CPD) for a tensor $\mathcal{T} \in \mathbb{C}^{N \times N \times N}$ is in the order of $\mathcal{O}\left(N^3 \log(N)\right)$, mainly due to the dominating multidimensional DFT.

5.4 Numerical experiments

To illustrate the performance of the presented methods, numerical experiments are conducted for both circulant and block-circulant factor matrices. In both experiments, the mean relative error (MRE) is used as a measure for accuracy. For an Nth-order tensor with factors $\mathbf{U}^{(n)}$, the MRE is defined as

$$\text{MRE} = \frac{1}{N} \sum_{n=1}^{N} \frac{\left|\left|\mathbf{U}^{(n)} - \widehat{\mathbf{U}}^{(n)} \Pi \Delta_n\right|\right|_{\text{F}}}{\left|\left|\mathbf{U}^{(n)}\right|\right|_{\text{F}}}.$$

In this definition, the matrices Δ_n and Π represent the optimal scaling and permutation matrix to deal with the indeterminacies of the CPD as discussed in Section 5.2.

All tensor computations are implemented using Tensorlab and all optimization-based decompositions are computed using a nonlinear least squares algorithm [113, 156].

5.4.1 Circulant factors

Consider a tensor $\mathcal{T} \in \mathbb{C}^{3 \times 3 \times 3}$ with rank 2. It is constructed such that its factor matrices $\mathbf{U}^{(n)} \in \mathbb{C}^{3 \times 2}$ have circulant structure. The generating elements of the different factors have been randomly sampled from a standard normal distribution. After construction of the tensor, additive Gaussian noise was added to obtain a predefined signal-to-noise ratio. The noisy tensor is then decomposed using various methods and the accuracy of the estimated factors is reported.

The mean results of 100 experiments are shown in Table 5.1. The method presented in [63] seems to perform worst. Note that this experiment is essentially the illustrative example given in [63], where the authors have stated that their

Table 5.1: Our method outperforms a naive implementation of [63] and is nearly optimal when compared to a full optimization-based solution. The table shows the mean estimation error over 100 experiments expressed in dB for various SNRs.

	SNR			
	0 dB	10 dB	20 dB	30 dB
Method from [63]	−3.84	−6.14	−11.92	−18.24
Method from this paper	−9.72	−20.36	−30.50	−40.73
Optimization initialized with this paper's method	−11.47	−21.92	−31.84	−41.81

way of solving the set of monomial equations is not guaranteed to be optimal in any sense. If more equations are taken into account, the accuracy is expected to improve until it matches the accuracy of our method, which automatically takes all available information into account. In this case, the tensor $\mathcal{A}$ in (5.4) has no zero entries, which allows us to compute the CPD of $\mathcal{G}$ using a generalized eigenvalue decomposition. This implies that the presented algorithm is fully algebraic in this experiment. If this algebraic result is used to initialize a fully-structured optimization-based method based on nonlinear least squares, the results are similar. This implies that the algebraic method is already quite accurate.

5.4.2 Block-circulant factors

Consider a tensor $\mathcal{T} \in \mathbb{C}^{9\times 9\times 9}$ with rank 9. Its factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ have block-circulant structure with block width $R = 3$. More specifically, $\mathbf{A}$ and $\mathbf{B}$ are block-diagonal matrices with blocks of size 3×3. The diagonal blocks within each factor are all the same since the matrices have block-circulant structure. Their entries have been randomly sampled from a standard normal distribution. The factor $\mathbf{C}$ has block-circulant structure as depicted in Figure 5.1 with shifts of size $M_c = 2$. Its generating elements are randomly selected from a standard normal distribution. The noisy tensor is obtained by adding Gaussian noise to the exact tensor.

The mean results of 500 experiments are presented in Figure 5.2. Because of the specific construction of $\mathbf{A}$ and $\mathbf{B}$, the matrix $\mathbf{B} \odot \mathbf{A}$ has block-Toeplitz structure, which allows us to compute the factors using the subspace-based method from [147]. This method exploits the block-Toeplitz structure in two

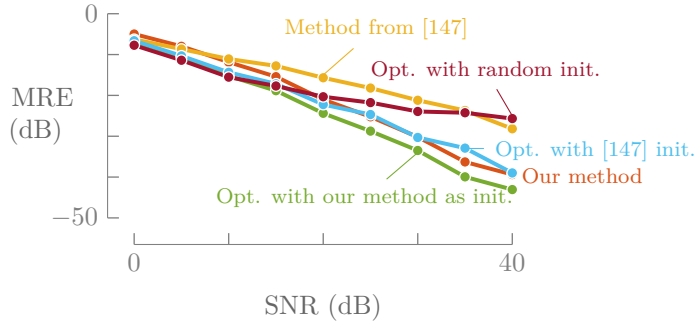


Figure 5.2: The method presented in this paper outperforms other methods, both as standalone method and as initialization for optimization-based methods. The figure shows the mean accuracy over 500 experiments.

Table 5.2: Execution times of various algorithms.

	Execution time (s)
Our method	0.093 s
Method from [147]	0.075 s
Optimization-based method, init. with our method	3.54 s
Optimization-based method, init. with [147]	4.45 s
Optimization-based method, random init.	46.62 s

modes to find two factors. The third factor can then be computed using the estimates of the other factors.

In our proposed method, 22% of the entries of $\mathcal{A}$ are zero, which implies that 22% of the entries of $\mathcal{G}$ are missing. However, there are six frontal slices that do not contain any missing entries. We use a GEVD of two of these known slices to partially initialize a nonlinear least squares method that computes the full decomposition of $\mathcal{G}$ [156].

The fully structured optimization-based method constrains the factors to have block-circulant structure. The initialization of this method is either random, is done using our presented method, or is done using the method from [147]. All three initialization strategies are shown in Figure 5.2. For the random initialization, the algorithm was run three times and the best result was retained.

Figure 5.2 shows that the subspace-based algorithm from [147] is the least accurate for most SNR values. This is because the structure in only two modes is exploited, whereas the other methods take all structure into account. Our

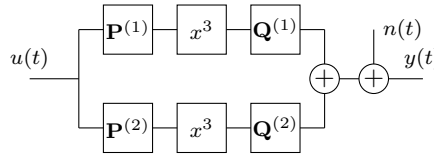


Figure 5.3: Block scheme of a parallel Wiener–Hammerstein system.

proposed algorithm is especially accurate for higher SNRs, where it outperforms a fully structured decomposition with random initialization and matches the optimization-based method initialized by [147]. The best result is achieved when our proposed method is used to initialize the fully structured decomposition. In the previous experiment, there was hardly any difference between our proposed algorithm and the fully structured one. In this experiment, the difference is larger because the missing elements reduce the available information. Computation times of all methods are shown in Table 5.2 and were measured at an SNR of 20 dB on a standard laptop containing 16 GB of RAM and an Intel Core i7-4810MQ processor. The results show that the speed of our method and that of [147] is comparable. However, for tensors with all (block-)circulant factors, our method is more accurate. The presented method is thus a valuable standalone and initialization algorithm.

5.4.3 Parallel Wiener–Hammerstein system identification

The identification of a homogeneous Wiener–Hammerstein system can be formulated as a CPD with almost all circulant factors [63, 77, 57]. A Wiener–Hammerstein system is defined as a static nonlinear block sandwiched between two linear time-invariant FIR filters and can be identified by first estimating a Volterra kernel from input/output data and then computing its CPD. This idea was extended to parallel Wiener–Hammerstein systems in [56]. It is shown that the estimated Volterra kernels admit a structured tensor decomposition with almost all block-circulant factors. An optimization-based strategy to solve this problem has been developed in [56]. Here, we focus on the special case depicted in Figure 5.3. If $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ are simple moving average filters of length $L+1$, it immediately follows from [56] that the third-order Volterra kernel $\mathcal{K}$ can be written as a CPD with pure block-circulant factors containing the filter coefficients of $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$. Note that for arbitrary filters $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$, one has to resort to optimization-based techniques such as [56]. How the parameters appear in the decomposition is explained in [56] as well.

As an illustration, we simulate the system shown in Figure 5.3. The input signal

$u(t)$ consists of 10^4 random samples from a standard normal distribution. The filters $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ consist of 6 samples from a standard normal distribution. The filters $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ are simple moving average filters of length 6. White Gaussian noise $n(t)$ is added to obtain an SNR of 20 dB. After estimating the third-order Volterra kernel as in [56], our method was used to retrieve the coefficients of $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$. Our method reaches an MRE of 0.021 whereas the method from [147] achieves an MRE of 0.049.

5.5 Conclusion

We presented a method to efficiently compute a CPD with block-circulant factors by building on the existing method from [63]. By transforming the data tensor to the frequency domain and computing an element-wise division, the rank of the tensor is reduced to the width of the circulant block-columns. The resulting (possible incomplete) tensor can then be decomposed using standard algebraic or optimization-based methods. This yields an easy-to-compute and accurate estimate that can be used directly or as initialization for optimization-based methods that take the full structure into account.

Chapter 6

Tensor similarity in two modes

Abstract — Multi-way datasets are widespread in signal processing and play an important role in blind signal separation, array processing and biomedical signal processing, among others. One key strength of tensors is that their decompositions are unique under mild conditions, which allows the recovery of features or source signals. In several applications, such as classification, we wish to compare factor matrices of the decompositions. Though this is possible by first computing the tensor decompositions and subsequently comparing the factors, these decompositions are often computationally expensive. In this paper, we present a similarity method that indicates whether the factors in two modes are essentially equal without explicitly computing them. Essential equality conditions, which ensure the theoretical validity of our approach, are provided for various underlying tensor decompositions. The developed algorithm provides a computationally efficient way to compare factors. The method is illustrated in a context of emitter movement detection and fluorescence data analysis.

Reference — This chapter is a slightly adapted version of [145]. The candidate did the research and wrote the article under the guidance of the coauthor. Part of the results of this chapter have been published in a book chapter as well aiming to introduce the concept of tensor similarity to the chemometrics community [144]. This book chapter can be found in Appendix A.

6.1 Introduction

The increasing digitalization of society and advances in data acquisition lead to a growing amount of multiway datasets. These datasets can be naturally stored in tensors, which are higher-order generalizations of vectors and matrices. Apart from offering natural data structures, higher-order tensors are useful tools for analysis as well. One favorable property is that their decompositions are unique under mild conditions, which contrasts with matrix decompositions that require additional constraints to attain uniqueness.

Initially driven by applications in psychometrics and chemometrics, tensors have made their entrance in signal processing and machine learning from the 90s onwards [81, 110]. A variety of applications in these domains have been tackled using tensors. In signal processing, tensors have proven to be extremely useful in blind signal separation problems, where the underlying sources are estimated from a set of mixtures. To solve these problems, additional assumptions have to be imposed on the sources. Examples include statistical independence [28], low-rank conditions [14], and many more [26, 46, 32]. Another example application in signal processing is emitter localization [109, 99]. In machine learning, tensors have been used in recommender systems and topic modeling (see [110] and references therein), face recognition [149], classification [25] and for learning latent variable models [7], to name a few.

Many of these applications rely on computing the factors of a tensor decomposition. However, in several applications, such as classification, we are interested in comparing the factors rather than in their explicit values. Because tensor decompositions are computationally expensive and can be ill-conditioned, strategies that are able to compare factors without explicitly computing them are of interest. In [33], a subspace-based method is derived that allows one to determine whether all factors of two tensors are equal without explicitly computing them. For third-order tensors for instance, this method can determine whether all three factor matrices are equal. However, in several applications the relevant information is contained in just two modes of a tensor. For instance, tensors used in emitter localization contain the location information in the first two modes and the (continuously changing) source information in the third mode [109, 14]. Consequently, we are only interested in the first two modes if we wish to verify whether the emitter locations have changed. Another example can be found in chemometrics, where third-order tensors contain excitation-emission information of chemical compounds [15, 112, 46]. If the goal is to check whether sets of mixtures consist of the same compounds, it suffices to compare the factors in the first two modes, as we will illustrate in Section 6.6.

In this paper, we develop a subspace-based method that is able to compare

two third-order tensors in just two modes without imposing strong constraints on the third mode. More explicitly, the factor matrices in the first two modes are compared up to trivial indeterminacies such as column permutation and scaling. Note that this can be interpreted as the comparison of latent variables without explicitly computing them. The difference with the method from [33] lies in specialization. In [33], the underlying decomposition of two tensors is unknown. Several subspaces of the tensors are then compared to determine which decomposition they admit and whether the terms are the same. In certain applications, the underlying decomposition is known. Our method exploits this prior knowledge, which allows one to check just one subspace equality to determine whether the factors in two modes are equal. We also provide the theoretical foundations that ensure the subspace comparisons are equivalent to comparing the factor matrices.

A few related papers can be found in the literature. For instance, the results of [33] have been used to develop tensorial kernels in [111]. Another related paper is [79], where tensor canonical correlation analysis (TCCA) is presented. In other domains, such as time series clustering, similarity between datasets has been researched in more detail. In [89], three types of similarity approaches are distinguished for time series: similarity based on the raw data, based on extracted features and based on data models. The similarity measure for tensors presented in this chapter conceptually fits in the latter category, in which the model is the tensor decomposition at hand.

As will be explained in Section 6.3, the similarity method represents the comparison of two full third-order tensors as one vector of principal angles. This resulting vector can then easily be used in classification algorithms such as support vector machines, neural networks or decision trees. In this sense, our method can be interpreted as a dimensionality reduction strategy. This allows us to classify entire tensors based on the similarity of their underlying decompositions in two modes.

The remainder of this section introduces the notations used throughout this text. In Section 6.2 we introduce the relevant tensor decompositions with their properties. The core idea of the method is subsequently introduced in Section 6.3, together with the algorithm and computational remarks. Sections 6.4 and 6.5 then provide the theoretical details for full and partial similarity. The results are illustrated by a numerical experiment and two applications in Section 6.6.

Notations

Scalars are represented by italic lowercase letters (e.g., x), column vectors by bold lowercase letters (e.g., $\mathbf{x}$), matrices by bold uppercase letters (e.g., $\mathbf{X}$) and

tensors by calligraphic letters (e.g., $\mathcal{X}$). Subscripts are used to denote subsets of vectors, matrices and tensors. For instance, a_n represents the nth entry of the vector $\mathbf{a}$ while $\mathbf{b}_n$ represents the nth column of the matrix $\mathbf{B}$ and t_{ijk} represents the entry with indices (i, j, k) of the tensor $\mathcal{T}$. The Kronecker product of two matrices $\mathbf{A} \in \mathbb{C}^{I \times J}$ and $\mathbf{B} \in \mathbb{C}^{K \times L}$ is denoted by $\mathbf{A} \otimes \mathbf{B} \in \mathbb{C}^{IK \times JL}$. The Khatri–Rao product of partitioned matrices $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_N]$ and $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_N]$ is given by

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{A}_1 \otimes \mathbf{B}_1, \ldots, \mathbf{A}_N \otimes \mathbf{B}_N] .$$

Unless explicitly stated otherwise, all partitionings in this paper will be column-wise (i.e., $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$). The outer product is denoted by $\circ$. The transpose and the conjugate transpose are denoted by $\cdot^{\mathrm{T}}$ and $\cdot^{\mathrm{H}}$, respectively. The column space of a matrix $\mathbf{A}$ is denoted by $\operatorname{span}\{\mathbf{A}\}$. Column-wise vectorization of a matrix is denoted by $\operatorname{Vec}\{\mathbf{A}\}$.

6.2 Tensor decompositions

We briefly review three relevant tensor decompositions: the canonical polyadic decomposition (CPD), the block term decomposition (BTD) in multilinear rank-$(L, L, 1)$ terms, and the BTD in multilinear rank-$(L, L, \cdot)$ terms.

6.2.1 Canonical polyadic decomposition

A rank-1 tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ of order N is defined as the outer product of N nonzero vectors $\mathbf{u}^{(n)} \in \mathbb{C}^{I_n}$, i.e., $\mathcal{A} = \mathbf{u}^{(1)} \circ \cdots \circ \mathbf{u}^{(N)}$. The rank of a tensor $\mathcal{T}$ is then defined as the minimum number of rank-1 terms that yield $\mathcal{T}$ in a linear combination. Such a (minimal) linear combination of rank-1 terms is called a canonical polyadic decomposition (CPD). Mathematically, the CPD of a rank-R tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ can be written as

$$\mathcal{T} = \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(N)}. \tag{6.1}$$

The factor vectors $\mathbf{u}_r^{(n)} \in \mathbb{C}^{I_n}$ are often concatenated in factor matrices $\mathbf{U}^{(n)} = \left[\mathbf{u}_1^{(n)}, \ldots, \mathbf{u}_R^{(n)}\right] \in \mathbb{C}^{I_n \times R}$. Following the notation of [81], (6.1) can be concisely written as

$$\mathcal{T} = \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right], \tag{6.2}$$

with $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R] \in \mathbb{C}^R$. A graphical representation of the CPD is shown in Figure 6.1.

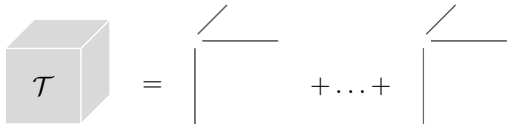


Figure 6.1: Graphical illustration of a CPD.

One of the key strengths of the CPD is its essential uniqueness under mild conditions. By essential uniqueness, we mean that the decomposition is unique up to two trivial indeterminacies. First, the different rank-1 terms can be arbitrarily permuted. Second, each vector within a rank-1 term can be scaled as long as the other vectors of the rank-1 term are counterscaled. Essential uniqueness conditions for CPDs have been extensively studied in literature, see e.g. [52, 50, 110]. Various algorithms to compute CPDs have been developed as well, see e.g. [110, 113] and references therein.

In this paper, we will mainly rely on uniqueness conditions of third-order tensors of which the third factor matrix has full column rank. Uniqueness conditions for this case are well-known and one set of conditions that we will use in this paper is given in Theorem 1. In this theorem, $\mathcal{C}_k(\mathbf{A})$ denotes the kth compound matrix of an $I \times R$ matrix $\mathbf{A}$, which is the $\binom{I}{k} \times \binom{R}{k}$ matrix containing the determinants of all $k \times k$ submatrices of $\mathbf{A}$ [49]. The determinants are arranged such that the submatrix index sets are in lexicographic order.

Theorem 1. *Consider the polyadic decomposition of $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ as defined in* (6.2)*. If*

$$\begin{cases} \mathbf{U}^{(3)} \in \mathbb{C}^{I_3 \times R} \text{ has full column rank,} \\ \mathcal{C}_2\left(\mathbf{U}^{(1)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(2)}\right) \text{ has full column rank,} \end{cases}$$

then the rank of $\mathcal{X}$ is R and the CPD of $\mathcal{X}$ is essentially unique [74, 30, 49, 50].

Let us consider the unfolding of a tensor to a matrix. A third order tensor $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \in \mathbb{C}^{I \times J \times K}$ can be unfolded in several ways. We will use $\mathbf{T}_{IJ \times K}$ to denote the matrix of which the kth column is the column-wise vectorization of the kth frontal slice of $\mathcal{T}$. This matrix can be written in terms of the CPD factors as follows:

$$\mathbf{T}_{IJ \times K} = (\mathbf{B} \odot \mathbf{A})\mathbf{C}^{\mathrm{T}} \in \mathbb{C}^{IJ \times K}.$$

Similar expressions exist for higher-order tensors and for unfoldings to other modes [81].



Figure 6.2: Graphical illustration of a BTD in multilinear rank-$(L, L, 1)$ terms.

6.2.2 Block term decomposition in rank-$(L, L, 1)$ terms

A block term decomposition (BTD) in multilinear rank-$(L, L, 1)$ terms is a generalization of a CPD that allows low-rank contributions in the first two modes of each term [31]. Mathematically, the BTD of a third-order tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ in rank-$(L, L, 1)$ terms can be written as

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

in which $\mathbf{A}_r \in \mathbb{C}^{I \times L}$, $\mathbf{B}_r \in \mathbb{C}^{J \times L}$ and $\mathbf{c}_r \in \mathbb{C}^{K}$. Note that a connection can be made to the PARALIND-decomposition [16]. A graphical representation of a BTD in rank-$(L, L, 1)$ is shown in Figure 6.2. Essential uniqueness conditions for this decomposition are given in [31, 32, 117]. The terms can again be arbitrarily permuted, but the scaling indeterminacy is slightly more complex than before. Within each term, one can postmultiply $\mathbf{A}_r$ by any nonsingular $\mathbf{G}_r \in \mathbb{C}^{L \times L}$ as long as $\mathbf{B}_r$ is postmultiplied by $\left(\mathbf{G}_r^{-1}\right)^{\mathrm{T}}$. Moreover, the factors within each rank-$(L, L, 1)$ term can be scaled and counterscaled as long as their product remains the same. Algorithms for this type of decomposition can be found in [113, 41, 16].

The matrix unfolding $\mathbf{T}_{IJ \times K}$ is given by

$$\mathbf{T}_{IJ \times K} = \left[(\mathbf{B}_1 \odot \mathbf{A}_1) \, \mathbf{1}_L, \ldots, (\mathbf{B}_R \odot \mathbf{A}_R) \, \mathbf{1}_L \right] \cdot \mathbf{C}^{\mathrm{T}}.$$

6.2.3 Block term decomposition in rank-$(L, L, \cdot)$ terms

Mathematically, the BTD in multilinear rank-$(L, L, \cdot)$ terms of a tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ is given by

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r,$$

in which $\mathcal{D}_r \in \mathbb{C}^{L \times L \times K}$ (with mode-1 rank equal to L and mode-2 rank equal to M) and in which $\mathbf{A}_r \in \mathbb{C}^{I \times L}$ and $\mathbf{B}_r \in \mathbb{C}^{J \times L}$ have full column rank. Note that the mode-n rank of a tensor is defined as the dimension of the vector space



Figure 6.3: Graphical illustration of a BTD in multilinear rank-$(L, L, \cdot)$ terms.

spanned by the mode-n vectors of the tensor. A graphical representation of this decomposition is given in Figure 6.3.

Again, essential uniqueness conditions can be found in [31]. The different terms can be arbitrarily permuted and the factor matrices $\mathbf{A}_r$ and $\mathbf{B}_r$ can be postmultiplied by nonsingular matrices $\mathbf{X}_r$ and $\mathbf{Y}_r$ respectively, as long as the core tensor $\mathcal{D}_r$ is replaced by $\mathcal{D}_r \bullet_1 \mathbf{X}_r^{-1} \bullet_2 \mathbf{Y}_r^{-1}$. Algorithms to compute this type of decomposition can for instance be found in [41, 114].

6.3 Problem statement and algorithm

6.3.1 Problem statement

As explained in the introduction, the comparison of tensors in the first two modes consists of verifying whether their factors in these modes are equal up to trivial indeterminacies. The factors will obviously depend on the type of decomposition the tensors admit. Let two tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{C}^{I \times J \times K}$ both admit the same type of decomposition, which may be a CPD, BTD in rank-$(L, L, 1)$ terms or a BTD in rank-$(L, L, \cdot)$ terms. In general, we can compactly write the tensor decompositions as follows:

$$\mathcal{T}^{(1)} = \left[\!\left[\mathcal{G}^{(1)}; \mathbf{A}, \mathbf{B}, \mathbf{C} \right]\!\right],$$

$$\mathcal{T}^{(2)} = \left[\!\left[\mathcal{G}^{(2)}; \mathbf{D}, \mathbf{E}, \mathbf{F} \right]\!\right],$$

in which the structure of the core tensors $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ depends on the underlying decomposition [25]. The details of this compact representation, such as the structure of the core tensors, can be found in [25, 31] but are not that important for our purposes. The main goal in this paper is to compare the matrices $\mathbf{A}$ and $\mathbf{D}$, and $\mathbf{B}$ and $\mathbf{E}$ up to the trivial indeterminacies associated with the underlying decompositions. This is represented graphically for the CPD case in Figure 6.4. Throughout the rest of this text, we will denote equality of the factors in two modes up to trivial indeterminacies by essential equality in two modes. For

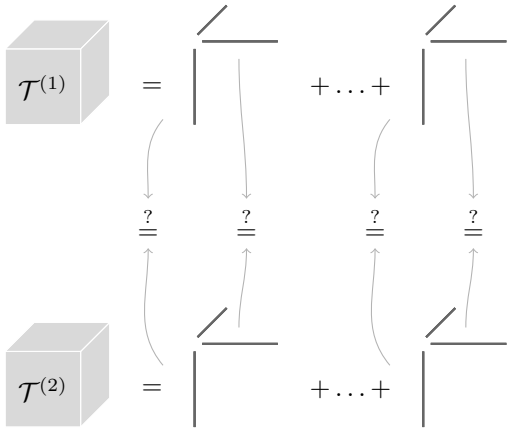


Figure 6.4: The goal of the paper is to verify whether the factor matrices of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ in the first two modes are equal without explicitly computing the decompositions and using only linear algebra, illustrated here for the case in which both tensors admit a CPD.

tensors that do not admit unique decompositions, essential equality means that there exists a pair of decompositions of the tensors that are essentially equal.

6.3.2 Algorithm overview

To check essential equality in two modes without explicitly computing the factors, we turn to subspace comparisons. More specifically, the remainder of the text will show that under certain decomposition-dependent conditions, $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{C}^{I \times J \times K}$ are essentially equal in two modes if and only if

$$\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}.$$

This calls for a measure to compare subspaces. One way to characterize subspace similarity consists of using a set of principal angles. To draw conclusions from these angles, further processing is often needed. This can be done by a variety of popular machine learning algorithms, ranging from simple thresholding to more advanced support vector machines and clustering.

The decomposition-dependent conditions for the method play a fundamental role, as they ensure that the underlying decompositions are essentially unique. This ensures that the compared factors are unique and allows the use of subspaces

to compare the underlying factor matrices. Further relaxation of the conditions to non-unique decompositions is briefly considered as well in Section 6.4.1.

A high-level overview of the procedure is given in Algorithm 5. The mathematical and computational details are provided in the subsequent sections. More specifically, the full mathematical proofs of the assumptions are given in Sections 6.4 and 6.5, and the computational details are discussed in Section 6.3.3. This presentation was chosen to highlight the simplicity and accessibility of the presented algorithm without clouding the view with mathematics.

Algorithm 5: High-level overview of the procedure to check tensor similarity in two modes.

Data: tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{C}^{I \times J \times K}$
Result: Decision whether $\mathbf{A} \stackrel{?}{=} \mathbf{D}$ and $\mathbf{B} \stackrel{?}{=} \mathbf{E}$

1. Check if all assumptions hold (in the generic case)
2. Unfold $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ to $\mathbf{T}^{(1)}_{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K}$
3. Select an appropriate value N for the number of components
4. Find basis $\mathbf{U} \in \mathbb{C}^{IJ \times N}$ for dominant span of $\mathbf{T}^{(1)}_{IJ \times K}$
5. Find basis $\mathbf{V} \in \mathbb{C}^{IJ \times N}$ for dominant span of $\mathbf{T}^{(2)}_{IJ \times K}$
6. Compute N principal angles between $\mathbf{U}$ and $\mathbf{V}$
7. Classify the principal angles

6.3.3 Computational remarks

In this section, additional information is provided for the various steps in Algorithm 5. These remarks should clarify how the method can be used in practice.

Verifying conditions

Theorems 2 to 6 in the following sections specify which conditions must hold such that Algorithm 5 yields unambiguous results. If the factor matrices are known, the verification of the assumptions is straightforward. However, the goal of our similarity method is to avoid computing the factors explicitly. Here, we

show that it is still possible to verify the conditions without explicitly computing the factors $\mathbf{A}, \mathbf{B}, \mathbf{D}$ and $\mathbf{E}$.

One of the recurring assumptions is a full-column-rank factor matrix. Take the example of a rank-R CPD of $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$, given by $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. To check whether $\mathbf{C} \in \mathbb{C}^{K \times R}$ has full column rank, one can consider the unfolded tensor $\mathbf{T} = (\mathbf{B} \odot \mathbf{A})\, \mathbf{C}^{\mathrm{T}} \in \mathbb{C}^{IJ \times K}$. By simply checking whether the rank of $\mathbf{T}$ equals R, we know that $\mathbf{C}$ must have full column rank as long as $R \leq K$ and $R \leq IJ$.

Considering the same tensor $\mathcal{T}$ defined above, another recurring condition that has to be checked is whether $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ has full column rank. Note that this is a $\binom{I}{2}\binom{J}{2} \times \binom{R}{2}$ matrix. To check whether this matrix has full column rank without computing $\mathbf{A}$ and $\mathbf{B}$, we use a result from [51] which transforms a CPD-admitting tensor to a matrix $\mathbf{Q}_2(\mathcal{T}) \in \mathbb{C}^{\binom{I}{2}\binom{J}{2} \times L}$ that can be written as

$$\mathbf{Q}_2(\mathcal{T}) = [\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})]\, \mathcal{Q}_2(\mathbf{C})^{\mathrm{T}}.$$

The details concerning the construction of $\mathbf{Q}_2(\mathcal{T})$ can be found in [51]. The important part for our purposes is that $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ is one of the factors of the matrix $\mathbf{Q}_2(\mathcal{T})$. By checking whether the rank of $\mathbf{Q}_2(\mathcal{T})$ equals $\binom{R}{2}$, we can determine whether $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ has full column rank, provided that $\binom{R}{2} \leq \binom{I}{2}\binom{J}{2}$ and $\binom{R}{2} \leq L$.

The previous approach checked the conditions deterministically. Alternatively, one can turn to generic conditions. For instance, a generic matrix is known to have full column rank if it has at least as many rows as columns. For the rank-R tensor $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \in \mathbb{C}^{I \times J \times K}$, this means that $\mathbf{C}$ generically has full column rank if $R \leq K$. Similarly, in [50] it is stated that the matrix $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ has full column rank in the generic case when it has at least as many rows as columns. Generic uniqueness conditions for essential uniqueness of a BTD in rank-$(L, L, 1)$ terms and a BTD in rank-$(L, L, \cdot)$ terms can be found in [31, 50].

Selecting an appropriate value for the number of components

Once the tensors have been reshaped to matrices $\mathbf{T}^{(1)}_{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K}$, the next step is to determine the number of (dominant) components N. Note that for underlying CPDs or BTDs in rank-$(L, L, 1)$ terms, N equals the number of terms R. However, this is not the case for a BTD in R rank-$(L, L, \cdot)$ terms, where the number of important components is given by $N = RL^2$. An appropriate value for N can either follow directly from prior knowledge of the application or can be estimated using linear algebra. The latter is done by checking the number of dominant singular values of $\mathbf{T}_{IJ \times K}$. Apart from some special cases, such as the

case where the number of components N is bigger than IJ or K, this approach gives a reasonable estimate for N.

If the number of components is incorrectly estimated, the principal angles will be affected. More specifically, if the estimated number of components N exceeds the actual number of components M, there will generally be M zero-valued principal angles and $N-M$ nonzero principal angles since the extra information only contains noise. If the estimated number of components N is lower than the actual number of components M, the situation is slightly more involved. Assume without loss of generality that $N = M - 1$. This implies that we only get a partial view of the full desired subspaces. Even if both partial subspaces stem from the same subspace, there will be nonzero principal angles because the partial subspaces do not contain exactly the same information. For example, if $N = 3$ and $M = 4$, there will be two zero-valued principal angles and 1 nonzero-valued principal angle. We will illustrate this point numerically in Section 6.6.

Constructing a basis for a dominant subspace

Given a subspace such as span $\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\}$, the goal is to construct an orthonormal basis for its N-dimensional dominant subspace. To do this, compute the left singular vectors of $\mathbf{T}^{(1)}_{IJ\times K}$ corresponding to the N dominant singular values. The dominant subspace can also be obtained by updating a previous subspace using tracking techniques (e.g., [162]). Once the basis vectors are found, store them as columns in a matrix $\mathbf{U} \in \mathbb{C}^{IJ\times N}$. Note that apart from the method based on singular value decompositions given here, there are many ways to find a basis for a subspace (see e.g., [95]).

Principal angles

One way to characterize subspace similarity consists of using principal angles. When two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{I\times N}$ are compared, the principal angles $\{\theta_1, \ldots, \theta_N\} \in [0, \pi/2]$ form a set of minimized angles between both subspaces. Mathematically, the principal angles can be defined recursively as (e.g., [11]):

$$\theta_i \stackrel{\text{def}}{=} \min\left\{\arccos\left(\frac{|\mathbf{a}^{\mathrm{T}}\mathbf{b}|}{||\mathbf{a}||\,||\mathbf{b}||}\right) \;\middle|\; \mathbf{a} \in \operatorname{span}\{\mathbf{A}\}, \ldots \right.$$

$$\left. \mathbf{b} \in \operatorname{span}\{\mathbf{B}\}, \mathbf{a} \perp \mathbf{a}_j, \mathbf{b} \perp \mathbf{b}_j \quad \forall j \in \{1, \ldots, i-1\}\right\},$$

in which $\mathbf{a}_j$ and $\mathbf{b}_j$ are the vectors corresponding to θ_j. Note that all principal angles are (approximately) zero if the subspaces are (approximately) equal. The definition also implies that if $\operatorname{span}\{\mathbf{A}\} \cap \operatorname{span}\{\mathbf{B}\}$ is Q-dimensional, there are exactly Q zero-valued principal angles. Computation of principal angles can for instance be done using the singular value decomposition, see the algorithms in [80]. Note that the principal angles can be used in noisy conditions as well. The effect of noise on the principal angles will be illustrated numerically in Section 6.6. By simply setting a threshold, similar tensors can still be identified by checking whether the principal angles are sufficiently small. The appropriate threshold depends on both the application and the noise level. Note that more advanced machine learning techniques than thresholding can be used as well. For instance, the principal angles can be fed into a neural network, support vector machine or decision tree to determine whether the factors are sufficiently similar.

6.4 Full equality

Full essential equality in two modes entails that we check whether the full factor matrices in two modes are essentially equal. Equality conditions are provided for various underlying decompositions of the tensors. More specifically, theorems are given for underlying CPDs, BTDs in rank-$(L, L, 1)$ terms and BTDs in rank-$(L, L, \cdot)$ terms.

6.4.1 CPD

Let two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)} \in \mathbb{C}^{I \times J \times K}$ admit a rank-R CPD:

$$\begin{aligned} \mathcal{T}^{(1)} &= [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] , \\ \mathcal{T}^{(2)} &= [\![\mathbf{D}, \mathbf{E}, \mathbf{F}]\!] . \end{aligned} \tag{6.3}$$

We now wish to assess when $\mathbf{A}$ and $\mathbf{D} \in \mathbb{C}^{I \times R}$, and $\mathbf{B}$ and $\mathbf{E} \in \mathbb{C}^{J \times R}$ are equal up to column scaling and permutation. Mathematically, we want to verify whether there exists an $s \in \{1, \ldots, R\}$ for each $r \in \{1, \ldots, R\}$ such that

$$\mathbf{a}_r \mathbf{b}_r^{\mathrm{T}} = \alpha_s \mathbf{d}_s \mathbf{e}_s^{\mathrm{T}},$$

in which α_s captures the possible scaling differences. Essential equality conditions are formulated in the following theorem:

Theorem 2. *Let two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}$ admit a rank-R CPD as in* (6.3). *If*

$$(i) \qquad \mathbf{C} \text{ and } \mathbf{F} \text{ have full column rank,}$$

$$(ii) \qquad \mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B}) \text{ has full column rank,} \tag{6.4a}$$

then $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes if and only if span $\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\}$ = span $\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$.

Proof. Because both $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ admit a CPD, their unfoldings can be written as

$$\mathbf{T}^{(1)}_{IJ\times K} = (\mathbf{B} \odot \mathbf{A})\,\mathbf{C}^{\mathrm{T}},$$

$$\mathbf{T}^{(2)}_{IJ\times K} = (\mathbf{E} \odot \mathbf{D})\,\mathbf{F}^{\mathrm{T}},$$

as explained in Section 6.2.1.

We first show that span $\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\}$ = span $\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$ is a necessary condition. Assume $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes, i.e., that $\mathbf{D} = \mathbf{A}\boldsymbol{\Lambda}^{(1)}\boldsymbol{\Pi}$ and $\mathbf{E} = \mathbf{B}\boldsymbol{\Lambda}^{(2)}\boldsymbol{\Pi}$ with $\boldsymbol{\Lambda}^{(1)}, \boldsymbol{\Lambda}^{(2)}$ diagonal scaling matrices and $\boldsymbol{\Pi}$ a permutation matrix. The matrix unfoldings of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ then become

$$\mathbf{T}^{(1)}_{IJ\times K} = (\mathbf{B} \odot \mathbf{A})\,\mathbf{C}^{\mathrm{T}},$$

$$\mathbf{T}^{(2)}_{IJ\times K} = (\mathbf{B} \odot \mathbf{A})\,\boldsymbol{\Lambda}^{(1)}\boldsymbol{\Lambda}^{(2)}\boldsymbol{\Pi}\mathbf{F}^{\mathrm{T}}.$$

Because $\mathbf{C}$ and $\mathbf{F}$ are assumed to have full column rank and $\boldsymbol{\Lambda}^{(1)}, \boldsymbol{\Lambda}^{(2)}, \boldsymbol{\Pi}$ are nonsingular, it immediately follows that span $\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\}$ = span $\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$.

Conversely, assume span $\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\}$ = span $\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$ holds to show that this subspace equality is also sufficient. Because $\mathbf{C}$ and $\mathbf{F}$ have full column rank, the subspace equality can be written as

$$(\mathbf{B} \odot \mathbf{A})\left(\mathbf{M}^{(1)}\right)^{\mathrm{T}} = (\mathbf{E} \odot \mathbf{D})\left(\mathbf{M}^{(2)}\right)^{\mathrm{T}}, \tag{6.5}$$

with $\mathbf{M}^{(1)}, \mathbf{M}^{(2)} \in \mathbb{C}^{R\times R}$ nonsingular matrices. Both sides of (6.5) represent an unfolded CPD of the same tensor. Consequently, we can write

$$\mathcal{G} = \left[\!\left[\mathbf{A}, \mathbf{B}, \mathbf{M}^{(1)}\right]\!\right] = \left[\!\left[\mathbf{D}, \mathbf{E}, \mathbf{M}^{(2)}\right]\!\right] \in \mathbb{C}^{I\times J\times R}. \tag{6.6}$$

It follows from Theorem 1 that the CPD in equation (6.6) is essentially unique since $\mathbf{M}^{(1)}$ is nonsingular and $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ has full column rank as assumed

in (6.4a). Consequently, the factor matrices $\mathbf{A}$ and $\mathbf{D}$, and $\mathbf{B}$ and $\mathbf{E}$ are equal up to the trivial indeterminacies stated in Section 6.2.1. □

Most results in this paper concern the comparison of factor matrices of tensors with essentially unique decompositions. Results for the non-unique case can be obtained as well. The following theorem shows that uniqueness of the CPDs of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ is not required to compare their factors in the first two modes.

Theorem 3. *Let two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}$ admit a (possibly non-unique) rank-R CPD as in* (6.3) *and let us construct the stacked tensor $\mathcal{T}^{(stack)} \in \mathbb{C}^{I \times J \times 2K}$ by concatenating $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ along the third mode. If $\mathcal{T}^{(stack)}$ admits a rank-R CPD, then $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes.*

Proof. If $\mathcal{T}^{(\text{stack})} \in \mathbb{C}^{I \times J \times 2K}$ admits a rank-R CPD, it can be written as

$$\mathcal{T}^{(\text{stack})} = \left[\!\!\left[\mathbf{U}, \mathbf{V}, \begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \end{bmatrix} \right]\!\!\right],$$

with $\mathbf{U} \in \mathbb{C}^{I \times R}$, $\mathbf{V} \in \mathbb{C}^{J \times R}$ and $\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \in \mathbb{C}^{K \times R}$. Since $\mathcal{T}^{(\text{stack})}$ is constructed by concatenating $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ along the third mode, it immediately follows that

$$\mathcal{T}^{(1)} = \left[\!\!\left[\mathbf{U}, \mathbf{V}, \mathbf{W}^{(1)} \right]\!\!\right],$$

$$\mathcal{T}^{(2)} = \left[\!\!\left[\mathbf{U}, \mathbf{V}, \mathbf{W}^{(2)} \right]\!\!\right].$$

These expressions show that there exist rank-R decompositions of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ with essentially equal factor matrices in the first two modes. □

6.4.2 BTD in rank-$(L, L, 1)$ terms

Let two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)} \in \mathbb{C}^{I \times J \times K}$ admit a BTD in rank-$(L, L, 1)$ terms:

$$\mathcal{T}^{(1)} = \sum_{r=1}^{R} \left(\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}} \right) \circ \mathbf{c}_r, \tag{6.7}$$

$$\mathcal{T}^{(2)} = \sum_{r=1}^{R} \left(\mathbf{D}_r \mathbf{E}_r^{\mathrm{T}} \right) \circ \mathbf{f}_r, \tag{6.8}$$

with $\mathbf{A}_r, \mathbf{D}_r \in \mathbb{C}^{I \times L}$ and $\mathbf{B}_r, \mathbf{E}_r \in \mathbb{C}^{J \times L}$.

We now wish to assess when $\mathbf{A}_r$ and $\mathbf{D}_r$, and $\mathbf{B}_r$ and $\mathbf{E}_r$ are equal for all values of r up to the trivial indeterminacies associated with the BTD in rank-$(L, L, 1)$ terms. We again denote this by essential equality in the first two modes. Mathematically, we want to verify whether there exists an $s \in \{1, \ldots, R\}$ for each $r \in \{1, \ldots, R\}$ such that

$$\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}} = \alpha_s \mathbf{D}_s \mathbf{E}_s^{\mathrm{T}},$$

in which α_s captures the possible scaling differences. Equality conditions for this case are formulated in the following theorem:

Theorem 4. *Consider two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}$ admitting a BTD in rank-$(L, L, 1)$ terms as defined in* (6.7) *and* (6.8)*. If following conditions hold*

(i) $\mathbf{C}$ *and* $\mathbf{F}$ *have full column rank,*

(ii) *The BTD* $\mathcal{G} = \sum_{r=1}^{R} \left(\mathbf{A}_r \mathbf{B}_r^T\right) \circ \mathbf{m}_r^{(1)}$ *from* (6.23)

is essentially unique, (6.9a)

then $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes if and only if span $\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = span\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$.

Proof. The proof is conceptually similar to the proof of Theorem 2 and is given in Appendix 6.8. □

6.4.3 BTD in rank-$(L, L, \cdot)$ terms

Let two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)} \in \mathbb{K}^{I\times J\times K}$ admit a BTD in rank-$(L, L, \cdot)$ terms:

$$\mathcal{T}^{(1)} = \sum_{r=1}^{R} \mathcal{S}_r^{(1)} \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r, \tag{6.10}$$

$$\mathcal{T}^{(2)} = \sum_{r=1}^{R} \mathcal{S}_r^{(2)} \bullet_1 \mathbf{C}_r \bullet_2 \mathbf{D}_r, \tag{6.11}$$

with $\mathbf{A}_r, \mathbf{D}_r \in \mathbb{K}^{I\times L}$ and $\mathbf{B}_r, \mathbf{E}_r \in \mathbb{K}^{J\times L}$. These tensors can be unfolded as

$$\mathbf{T}^{(1)}_{IJ\times K} = [\mathbf{B}_1 \otimes \mathbf{A}_1, \ldots, \mathbf{B}_R \otimes \mathbf{A}_R] \cdot \mathbf{S}^{(1)},$$

$$\mathbf{T}^{(2)}_{IJ\times K} = [\mathbf{D}_1 \otimes \mathbf{C}_1, \ldots, \mathbf{D}_R \otimes \mathbf{C}_R] \cdot \mathbf{S}^{(2)},$$

with $\mathbf{S}^{(1)}, \mathbf{S}^{(2)} \in \mathbb{K}^{RL^2 \times K}$ matrices consisting of the matricized versions of $\mathcal{S}_r^{(1)}$ and $\mathcal{S}_r^{(2)}$, respectively. The main question remains: when are $\mathbf{A}$ and $\mathbf{C}$, and $\mathbf{B}$ and $\mathbf{D}$ equal up to the trivial indeterminacies of a rank-$(L, L, \cdot)$ BTD. Conditions for this essential equality problem in two modes are given in

Theorem 5. *Consider two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}$ admitting a BTD in rank-$(L, L, \cdot)$ terms as defined in* (6.10) *and* (6.11)*. If following conditions hold*

$$(i) \qquad \mathbf{S}^{(1)} \text{ and } \mathbf{S}^{(2)} \text{ have full row rank,}$$

$$(ii) \qquad \text{The BTD } \mathcal{G} = \sum_{r=1}^{R} \mathcal{M}_r^{(1)} \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \text{ from (6.25)}$$

$$\text{is essentially unique,} \tag{6.12a}$$

then $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes if and only if $\text{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \text{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$.

Proof. The proof is conceptually similar to the proof of Theorem 2 and is given in Appendix 6.9. □

6.5 Partial equality: the CPD case

In the previous section, full essential equality in two modes was considered. Here, we discuss the case in which only a subset of the terms of two CPDs are essentially equal in two modes. In terms of the factors, this is the case where a subset of the factor vectors are essentially equal.

6.5.1 Verifying partial similarity

Let two tensors $\mathcal{T}^{(1)}, \mathcal{T}^{(2)} \in \mathbb{C}^{I\times J\times K}$ admit a rank-R CPD:

$$\begin{aligned} \mathcal{T}^{(1)} &= [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!], \\ \mathcal{T}^{(2)} &= [\![\mathbf{D}, \mathbf{E}, \mathbf{F}]\!], \end{aligned} \tag{6.13}$$

with $\mathbf{A}, \mathbf{D} \in C^{I\times R}$, $\mathbf{B}, \mathbf{E} \in C^{J\times R}$ and $\mathbf{C}, \mathbf{F} \in C^{K\times R}$. The following theorem shows that under certain conditions, the number of zero-valued principal angles equals the number of terms of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ that are essentially equal in two modes.

Theorem 6. *Consider two tensors* $\mathcal{T}^{(1)}, \mathcal{T}^{(2)} \in \mathbb{C}^{I \times J \times K}$ *admitting a rank-R CPD as defined in* (6.13)*. If*

$$(i) \qquad \mathbf{C} \text{ and } \mathbf{F} \text{ have full column rank } R, \tag{6.14a}$$

$$(ii) \qquad \mathcal{C}_2(\mathbf{B}) \odot \mathcal{C}_2(\mathbf{A}) \text{ has full column rank}, \tag{6.14b}$$

$$(iii) \qquad \mathcal{C}_2(\mathbf{E}) \odot \mathcal{C}_2(\mathbf{D}) \text{ has full column rank}, \tag{6.14c}$$

$$(iv) \qquad \text{One basis for span} \left\{ \mathbf{T}^{(1)}_{IJ \times K} \right\} \cap \text{span} \left\{ \mathbf{T}^{(2)}_{IJ \times K} \right\}$$
$$\text{admits a unique (unfolded) rank-}Q\text{ CPD}, \tag{6.14d}$$

then there are Q *essentially equal terms in two modes if and only if* Q *principal angles between the unfolded tensors* $\mathbf{T}^{(1)}_{IJ \times K}$ *and* $\mathbf{T}^{(2)}_{IJ \times K}$ *are zero, with* $Q \leq R$*.*

Proof. Consider the unfolded matrices $\mathbf{T}^{(1)}_{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K} \in \mathbb{C}^{IJ \times K}$ of the tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, which can be written as

$$\mathbf{T}^{(1)}_{IJ \times K} = (\mathbf{B} \odot \mathbf{A})\, \mathbf{C}^{\mathrm{T}}$$

$$\mathbf{T}^{(2)}_{IJ \times K} = (\mathbf{E} \odot \mathbf{D})\, \mathbf{F}^{\mathrm{T}}.$$

Since both $\mathbf{C}$ and $\mathbf{F}$ are assumed to have full column rank in (6.14a), it follows that $(\mathbf{B} \odot \mathbf{A})$ and $(\mathbf{E} \odot \mathbf{D})$ form bases for the column spaces of $\mathbf{T}^{(1)}_{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K}$, respectively.

$\Leftarrow$) Assume there are Q zero-valued principal angles. This implies that the common subspace $\text{span} \left\{ \mathbf{T}^{(1)}_{IJ \times K} \right\} \cap \text{span} \left\{ \mathbf{T}^{(2)}_{IJ \times K} \right\}$ is exactly Q-dimensional. Let the columns of $\mathbf{G} \in \mathbb{C}^{IJ \times Q}$ contain a basis for this subspace. We then have

$$\mathbf{G} = (\mathbf{B} \odot \mathbf{A})\, \mathbf{M}^{(1)} = (\mathbf{E} \odot \mathbf{D})\, \mathbf{M}^{(2)}, \tag{6.15}$$

in which $\mathbf{M}^{(1)}, \mathbf{M}^{(2)} \in \mathbb{C}^{R \times Q}$ have full column rank. Because we assume in (6.14d) that $\mathbf{G}$ admits an unfolded rank-Q CPD, we can write

$$\mathbf{G} = (\mathbf{V} \odot \mathbf{U})\, \mathbf{W}^{\mathrm{T}}, \tag{6.16}$$

with $\mathbf{U} \in \mathbb{C}^{I \times Q}$, $\mathbf{V} \in \mathbb{C}^{J \times Q}$ and $\mathbf{W} \in \mathbb{C}^{Q \times Q}$. Because $\mathbf{W}$ has full rank, we can use (6.15) and (6.16) to write

$$(\mathbf{V} \odot \mathbf{U}) = (\mathbf{B} \odot \mathbf{A})\, \mathbf{N}^{(1)} = (\mathbf{E} \odot \mathbf{D})\, \mathbf{N}^{(2)},$$

with $\mathbf{N}^{(1)} = \mathbf{M}^{(1)} \left(\mathbf{W}^{\mathrm{T}}\right)^{-1} \in \mathbb{C}^{R \times Q}$ and $\mathbf{N}^{(2)} = \mathbf{M}^{(2)} \left(\mathbf{W}^{\mathrm{T}}\right)^{-1} \in \mathbb{C}^{R \times Q}$. The qth column of $(\mathbf{V} \odot \mathbf{U})$ is given by

$$\mathbf{v}_q \otimes \mathbf{u}_q = \sum_{r=1}^{R} n_{rq}^{(1)} \left(\mathbf{b}_r \otimes \mathbf{a}_r\right) = \sum_{r=1}^{R} n_{rq}^{(2)} \left(\mathbf{e}_r \otimes \mathbf{d}_r\right).$$

A column-wise reshape of these equations to $I \times J$ matrices then gives

$$\mathbf{u}_q \circ \mathbf{v}_q = \sum_{r=1}^{R} n_{rq}^{(1)} \left(\mathbf{a}_r \circ \mathbf{b}_r\right) = \sum_{r=1}^{R} n_{rq}^{(2)} \left(\mathbf{d}_r \circ \mathbf{e}_r\right). \tag{6.17}$$

Conditions (6.14b) and (6.14c) are sufficient for the CPD uniqueness of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ when (6.14a) holds. If these conditions are satisfied, then following necessary and sufficient conditions hold as well [30]:

(i) $\mathbf{C}$ and $\mathbf{F}$ have full column rank, (6.18a)

$(\cdot)$ For every $\mathbf{w}$ that has at least two nonzero entries

we have

(ii) $\operatorname{rank}\left(\mathbf{K}\left(\mathbf{w}\right)\right) > 1$ and (6.18b)

(iii) $\operatorname{rank}\left(\mathbf{L}\left(\mathbf{w}\right)\right) > 1$, (6.18c)

in which (6.18a) is the same condition as (6.14a) and the functions $\mathbf{K}\left(\mathbf{w}\right)$ and $\mathbf{L}\left(\mathbf{w}\right)$ are defined as $\mathbf{K}\left(\mathbf{w}\right) = \sum_{r=1}^{R} w_r \left(\mathbf{a}_r \circ \mathbf{b}_r\right)$ and $\mathbf{L}\left(\mathbf{w}\right) = \sum_{r=1}^{R} w_r \left(\mathbf{d}_r \circ \mathbf{e}_r\right)$. It immediately follows from conditions (6.18b) and (6.18c) that equation (6.17) can never be satisfied if the linear combinations are nontrivial, i.e., if there is more than one term in the linear combination. Consequently, for each $q \in \{1, \ldots, Q\}$ we have

$$\mathbf{v}_q \otimes \mathbf{u}_q = n_{rq}^{(1)} \left(\mathbf{b}_r \otimes \mathbf{a}_r\right) = n_{sq}^{(2)} \left(\mathbf{e}_s \otimes \mathbf{d}_s\right),$$

for certain $r, s \in \{1, \ldots, R\}$. We thus have Q columns of $(\mathbf{B} \odot \mathbf{A})$ and $(\mathbf{E} \odot \mathbf{D})$ that are equal up to scaling, which implies that there are Q essentially equal terms in the first two modes.

$\Rightarrow$) Conversely, assume that exactly Q terms are essentially equal in the first two modes. In this case, the bases can be written as

$$\mathbf{B} \odot \mathbf{A} = \left[\mathbf{B}_{\mathrm{eq}} \odot \mathbf{A}_{\mathrm{eq}}, \mathbf{B}_{\mathrm{diff}} \odot \mathbf{A}_{\mathrm{diff}}\right] \tag{6.19}$$

$$\mathbf{E} \odot \mathbf{D} = \left[\left(\mathbf{B}_{\mathrm{eq}} \odot \mathbf{A}_{\mathrm{eq}}\right) \mathbf{\Lambda\Pi}, \mathbf{E}_{\mathrm{diff}} \odot \mathbf{D}_{\mathrm{diff}}\right], \tag{6.20}$$

in which the equal parts of the factor matrices are denoted by $\mathbf{A}_{\text{eq}} \in \mathbb{C}^{I \times Q}$ and $\mathbf{B}_{\text{eq}} \in \mathbb{C}^{J \times Q}$. The $Q \times Q$ matrices $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ are used to express possible column scaling and permutation, respectively. Because the principal angles are defined as a set of minimized angles between subspaces, it follows immediately from (6.19) and (6.20) that there will be at least Q zero-valued principal angles between the subspace of $\mathbf{T}^{(1)}_{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K}$. We now prove by contradiction that there are no more than Q zero-valued principal angles. Assume there are $Q + S$ zero-valued principal angles, with $S > 0$. This implies that the null space of $[\mathbf{B} \odot \mathbf{A}, \mathbf{E} \odot \mathbf{D}]$ is $(Q + S)$-dimensional. Note that this null space equals $\text{span}\left\{\mathbf{T}^{(1)}_{IJ \times K}\right\} \cap \text{span}\left\{\mathbf{T}^{(2)}_{IJ \times K}\right\}$. Because assumption (6.14d) states that the null space admits an unfolded rank-Q CPD, this situation cannot occur. We can conclude that under the assumptions (6.14a) to (6.14d) there will be exactly Q zero-valued principal angles when Q terms are essentially equal in the first two modes. □

Note that in the conditions and proof of Theorem 6, verifying whether a matrix space admits a unique CPD without explicitly computing the factors can be done using the techniques mentioned in 6.3.3.

6.5.2 Computing the common terms

When two tensors are partially equal in two modes, the common terms can be extracted without having to compute the other terms. This approach is especially advantageous when there are just a few common terms. The procedure to compute these common terms is given below.

First, compute the R-dimensional dominant column spaces of $\mathbf{T}^{(1)}_{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K}$ and store them in the matrices $\mathbf{T}^{(1)}_{\text{dom}}$ and $\mathbf{T}^{(2)}_{\text{dom}}$. These matrices form a basis for $(\mathbf{B} \odot \mathbf{A})$ and $(\mathbf{E} \odot \mathbf{D})$, respectively. Next, construct the matrix $\mathbf{Z} = \left[\mathbf{T}^{(1)}_{\text{dom}}, \mathbf{T}^{(2)}_{\text{dom}}\right] \in \mathbb{K}^{IJ \times 2R}$. Assuming without loss of generality that there are U common terms in the first two modes, let the columns of $\mathbf{N} \in \mathbb{C}^{2R \times U}$ form a basis for the null space of $\mathbf{Z}$. This matrix $\mathbf{N}$ can then be partitioned as

$$\mathbf{N} = \begin{bmatrix} \mathbf{R} \\ \mathbf{S} \end{bmatrix} \in \mathbb{K}^{2R \times U},$$

in which $\mathbf{R}, \mathbf{S} \in \mathbb{K}^{R \times U}$. Now construct the matrix $\mathbf{W}$ as

$$\mathbf{W} = \mathbf{T}^{(1)}_{\text{dom}} \mathbf{R} = -\mathbf{T}^{(2)}_{\text{dom}} \mathbf{S}.$$

This matrix contains a basis for the subspace spanned by the common columns of $(\mathbf{B} \odot \mathbf{A})$ and $(\mathbf{E} \odot \mathbf{D})$. Consequently, it holds that

$$\mathbf{W} = (\mathbf{B}_{\text{comm}} \odot \mathbf{A}_{\text{comm}})\, \mathbf{M}^{\mathrm{T}}, \tag{6.21}$$

in which $\mathbf{M} \in \mathbb{C}^{U \times U}$ is a nonsingular matrix, and $\mathbf{A}_{\text{comm}} \in \mathbb{C}^{I \times U}$ and $\mathbf{B}_{\text{comm}} \in \mathbb{C}^{J \times U}$ denote the common subsets of $\mathbf{A}$ and $\mathbf{D}$, and $\mathbf{B}$ and $\mathbf{E}$, respectively. Computation of the (unfolded) CPD of $\mathbf{W}$ in (6.21) then yields the columns of $\mathbf{A}$ and $\mathbf{B}$ of the common terms of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$.

6.5.3 Computing the distinct terms

The previous section considered how the common terms can be extracted from the subspaces without computing the full decompositions. However, difficulties arise when trying to extract terms that are distinct. A full treatment of ways to compute these distinct terms is outside the scope of the paper, but we briefly mention one approach that improves upon simply computing both CPDs. Start by extracting the common terms as described in the previous section. Next, use this result as partial initialization for the computation of the full CPDs. This may result in faster convergence for optimization-based methods.

6.6 Numerical experiments and applications

The developed algorithm will be illustrated using both synthetic and real-life data. First, the influence of noise on the principal angles is briefly discussed and the results are compared with a method that explicitly computes the factors. Next, the method is applied to emitter movement detection and fluorescence spectroscopy.

6.6.1 Influence of noise

In Section 6.3, we stated that the principal angles can still be used in noisy conditions. Using numerical simulations, we show how they are affected when noise is present.

Let two tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{R}^{10\times10\times10}$ admit a rank-4 CPD. The tensors are constructed as

$$\mathcal{T}^{(1)} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!],$$

$$\mathcal{T}^{(2)} = [\![\mathbf{A}\boldsymbol{\Gamma}, \mathbf{B}\boldsymbol{\Lambda}, \mathbf{F}]\!],$$

in which the entries of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{F} \in \mathbb{R}^{10\times4}$ are pseudorandomly drawn from a standard normal distribution. The matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times4}$ are diagonal with diagonal entries also drawn from a standard normal distribution. These tensors are clearly essentially equal in the first two modes. We now add Gaussian noise to both tensors such that various signal-to-noise ratios (SNR) are obtained. At each noise level, the principal angles are computed following the procedure outlined in Section 6.3.

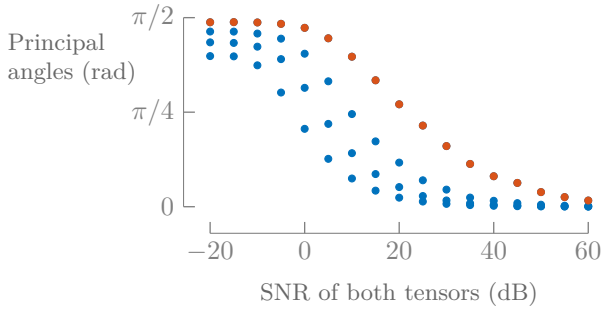
Figure 6.5a shows the mean principal angles between the unfolded tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ over 1000 experiments at each noise level. It is clear that the principal angles become smaller if the noise level decreases. This behavior matches the error on the factor matrices if the decompositions would be explicitly computed, as illustrated in Figure 6.6. For the CPD computation, the default nonlinear least-squares algorithm of Tensorlab was used [156].

By contrast, Figure 6.5b shows the principal angles between the unfoldings of two randomly generated $10 \times 10 \times 10$ tensors admitting a rank-4 CPD. The figure shows that all principal angles are large if the underlying tensors are not essentially equal in two modes, no matter what the noise level is. Again, this matches the expected behavior of the method that explicitly decomposes the tensors. Comparing figures 6.5a and 6.5b, we can conclude that the principal angles are able to distinguish between essentially equal and unequal tensors if the noise level is not too high.

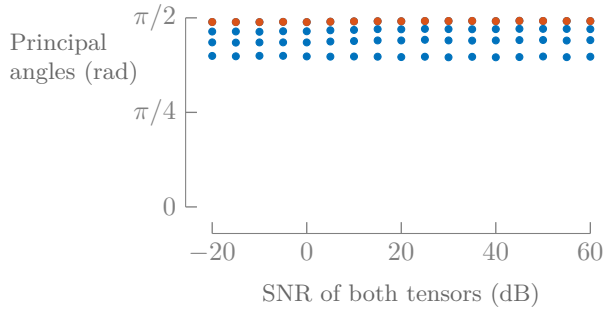
One of the main merits of our similarity method is its speed. Figure 6.7 shows that the similarity method is about two orders of magnitude faster than the approach that explicitly computes the factor matrices. Note that the speed of the latter depends on the noise level because the convergence of the CPD algorithm is generally slower in the presence of noise. By contrast, the speed of our similarity method is not affected. Similarly, our method is unaffected by the condition of the problem.

6.6.2 Incorrectly estimated number of components

In Section 6.3.3, we stated that the principal angles were affected if the number of components is incorrectly estimated. Using numerical simulations, we show how they are affected in a few examples.



(a) The principal angles are small when the underlying tensors are essentially equal in two modes and little noise is present.



(b) The principal angles are systematically large when the underlying tensors are not essentially equal.

Figure 6.5: Effects on the principal angles of varying noise levels on tensors that are essentially equal in two modes (a) and not essentially equal in two modes (b). For visual clarity, the largest principal value is shown in red.

Let two tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{R}^{10\times10\times10}$ admit a rank-4 CPD. The tensors are constructed as

$$\mathcal{T}^{(1)} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!],$$

$$\mathcal{T}^{(2)} = [\![\mathbf{A}\boldsymbol{\Gamma}, \mathbf{B}\boldsymbol{\Lambda}, \mathbf{F}]\!],$$

in which the entries of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{F} \in \mathbb{R}^{10\times4}$ are pseudorandomly drawn from a standard normal distribution. The matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times4}$ are diagonal with diagonal entries also drawn from a standard normal distribution. These tensors are clearly essentially equal in the first two modes. We now add Gaussian noise to both tensors such that various signal-to-noise ratios (SNR) are obtained. At each noise level, the principal angles are computed following the procedure

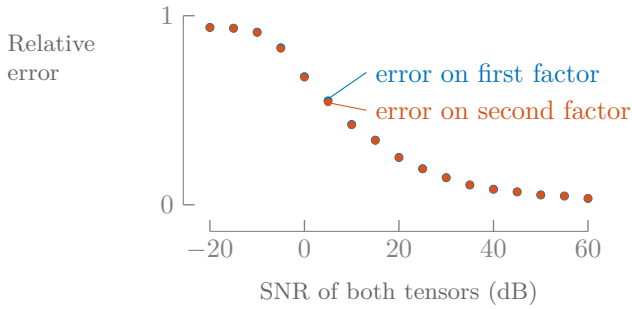


Figure 6.6: The relative errors on the first two factor matrices when the CPDs are explicitly computed show similar behavior as the principal angles. Note that the errors on both factors almost coincide.

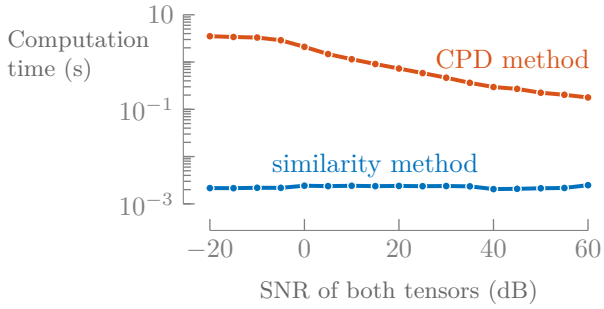
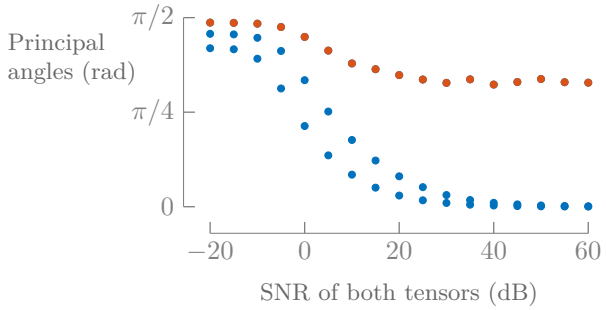


Figure 6.7: The similarity method is orders of magnitude faster than the method computing the CPDs and is independent of the noise level.

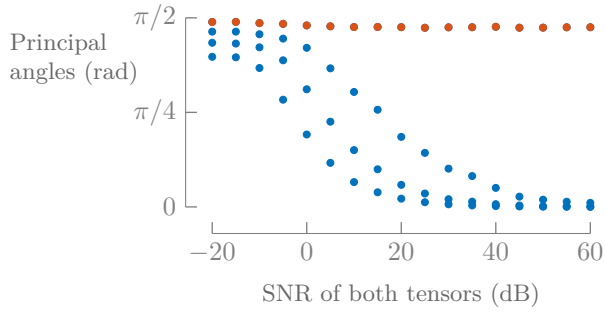
outlined in Section 6.3, but with the estimated number of components equal to $N = 3$.

Figure 6.8a shows the mean results of 300 experiments at each noise level, which clearly indicate that one of the principal angles remains nonzero. This is because the available 3-dimensional subspaces only contain a part of the actual 4-dimensional subspace, and those parts are not exactly the same. In this example, two principal angles are approximately zero for high SNRs, whereas the one that captures the differences between the partial subspaces remains large.

Similarly, let two tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{R}^{10\times10\times10}$ admit a rank-3 CPD. The tensors are constructed in exactly the same way as before, with the exeption that the number of components is different now. The principal angles are computed



(a) The estimated number of components is lower (3) than the actual number of components (4).



(b) The estimated number of components is higher (4) than the actual number of components (3).

Figure 6.8: Effects on the principal angles of an incorrectly estimated number of components, which is either chosen too low (a) or too high (b) for tensors that are essentially equal in two modes. For visual clarity, the largest principal value is shown in red.

following the procedure outlined in Section 6.3, but with the estimated number of components equal to $N = 4$, so the number of components is overestimated by one. The mean results of 300 experiments at each noise level are shown in Figure 6.8b. As explained in Section 6.3.3, the fourth principal angle does not involve subspace information, which is fully contained in the first three angles. Therefore, the fourth one only involves noise and is approximately equal to $\pi/2$.

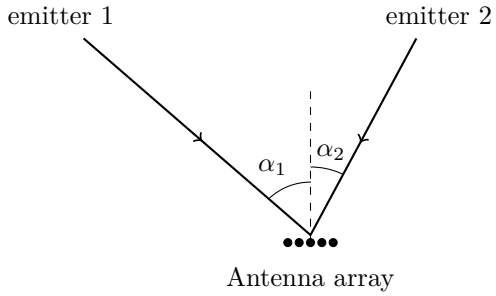


Figure 6.9: Direction of arrival estimation tries to retrieve the values α_r using solely the information obtained at the antenna array.

6.6.3 Application: Emitter movement detection

One possible application of our similarity measure can be found in emitter localization. Using an antenna array that receives data from several emitters, various methods are able to pinpoint the direction from which each emitter is transmitting, called the direction of arrival (DOA) [82, 125, 106, 105]. The direction-of-arrival problem is illustrated in Figure 6.9. Knowledge of this direction may be useful in military applications such as passive radar or sonar, or in telecommunications.

In [109], a tensor-based method is presented that captures the location information in the first two modes of a third-order tensor. We briefly repeat this method to clarify the relation between tensor decompositions and direction-of-arrival estimation. Consider a uniform linear array (ULA) of N uniformly spaced omnidirectional antennas as depicted in Figure 6.9. If this array receives signals from R narrowband emitters located in the far field, the received mixture $\mathbf{x}(t) \in \mathbb{C}^N$ at time instance t is given by

$$\mathbf{x}(t) = \mathbf{M}\mathbf{s}(t),$$

in which $\mathbf{s}(t) \in \mathbb{C}^R$ contains the emitter signals and $\mathbf{M} \in \mathbb{C}^{N \times R}$ is the mixing matrix with $m_{nr} = \theta_r^{n-1}$ and $\theta_r = e^{-2\pi i \Delta \sin(\alpha_r)\lambda^{-1}}$. Δ is the known inter-element spacing of the ULA, λ is the wavelength, and the angle α_r to the normal is the direction of arrival of the rth emitter and lies between $-\pi/2$ and $\pi/2$. It is shown in [109] that by rearranging the received mixture $\mathbf{x}$(t) into a matrix and stacking the obtained matrices at several time instances t, a tensor is obtained that admits a CPD. The factors in the first two modes contain the mixture information and the factor in the third mode contains the emitter data information. Using the first two factor matrices the matrix $\mathbf{M}$ can

be reconstructed, from which the directions of arrival α_r can be extracted. A similar approach for large-scale applications has been studied in [14].

Continuously monitoring the emitter locations requires the recomputation of the DOAs at each time step. However, this approach is computationally expensive and is cumbersome if emitters remain stationary for long time intervals. Moreover, this expensive computation also implies higher power usage, which may be an important issue for (remote) battery-driven computation units. By using our similarity method, one can use just a fraction of the computational effort to check whether any emitter has moved. When this is the case, the DOAs can be recomputed using existing techniques.

For the experiment, an ULA of 100 antennas was simulated receiving data from two emitters that move over time. The transmitted data consist of multi-modulus samples with continuous phase shift keying and are constructed by randomly sampling from a uniform distribution over concentric circles in the complex plane with radii 1, 2 or 3. Gaussian noise was added to the antenna outputs such that an SNR of 20 dB was obtained. In Figure 6.10, results for the simulated direction of arrival problem are shown. The top figure shows the true direction angles of the emitters. The middle figure then shows the estimated direction angles. This estimation is done using a sliding window of length 10 as follows. First, the method from [109] is used to estimate the starting angles using the first 10 samples. In doing so, a tensor is constructed by a simple reshaping operation, which is stored in $\mathcal{A}$. Next, the sliding window is moved one time instance and a new tensor $\mathcal{B}$ is constructed using the same reshaping procedure as in [109]. However, this tensor is not decomposed, but the similarity method is used to check whether $\mathcal{A}$ and $\mathcal{B}$ are essentially equal in the first two modes. If this is the case, the previously computed direction angles are retained. If at least one of the principal angles is different from zero, the new direction angles are computed by decomposing $\mathcal{B}$ and we replace $\mathcal{A}$ by $\mathcal{B}$ such that all subsequent tensors are compared to the one that has been decomposed last. In this experiment, we consider a principal angle to be different from zero if it is larger than 0.2 rad to take noise effects into account. The principal angles computed using this procedure are shown at the bottom of Figure 6.10. The vertical lines represent time instances at which the direction angles have been explicitly recomputed. The figure shows that this only happens when one of the emitters has moved, which is exactly the goal of this approach.

On average, checking the principal angles takes about $3.65 \cdot 10^{-4}$ s at each time instance, whereas recomputing the direction angles takes 0.0649 s, which is a difference of two orders of magnitude.

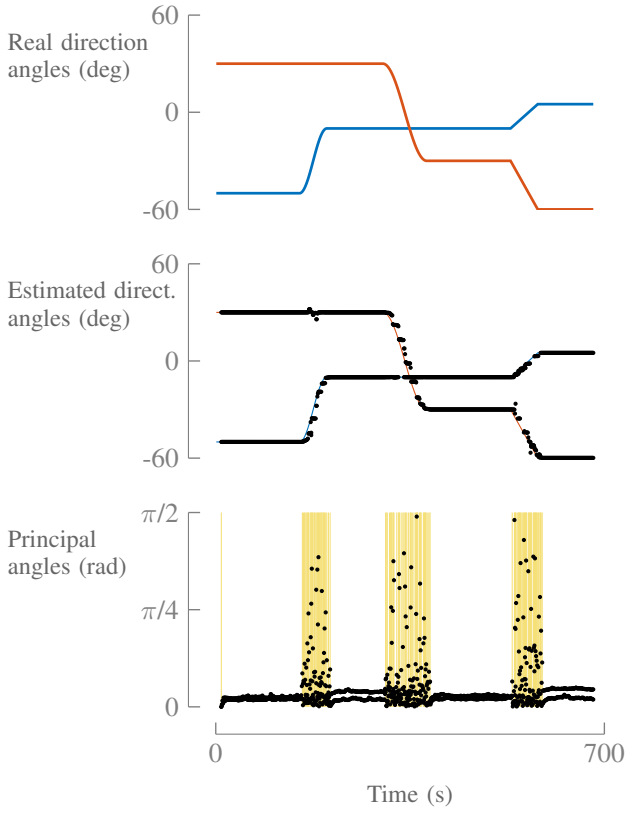


Figure 6.10: Using essential equality in two modes one can detect whether any emitter has displaced, which leads to lighter computational loads when there is no displacement.

6.6.4 Application: Fluorescence spectrography

Fluorescence spectroscopy data has been extensively analyzed in literature, often in an attempt to retrieve the pure analyte spectra making up the mixture, see [15] and references therein. The traditional way to do this involves stacking *several* excitation-emission matrices $\mathbf{E} \in \mathbb{C}^{K \times N}$, which gives a tensor admitting a low-rank CPD, see [15, 112] and references therein. The excitation-emission matrices are obtained by exciting the mixtures at K different excitation wavelengths and measuring the spectrum of the emitted light at N different emission wavelengths.

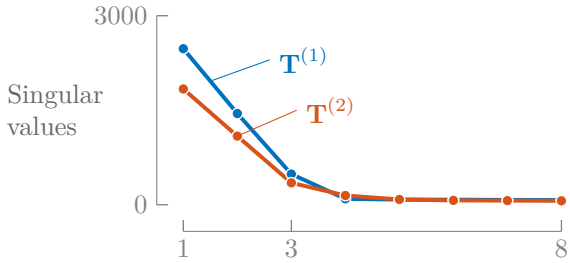


Figure 6.11: The first 3 singular values of $\mathbf{T}^{(1)}_{10100\times 61}$ and $\mathbf{T}^{(2)}_{10100\times 61}$ contain most of the information.

The real-life dataset[1] that we will use consists of two samples containing different relative concentrations of tyrosine (Tyr), tryptophane (Trp) and phenylalanine (Phe). Apart from these mixtures, measured excitation-emission spectra of the pure analytes are available as well. The excitation and emission wavelengths are $240-300$ nm and $250-450$ nm respectively, measured in steps of 1 nm. The obtained excitation-emission matrices thus have dimensions 201×61. A more detailed description of the data can be found in [15]. Here, we transform a *single* excitation-emission matrix to a tensor using Löwnerization [46]. In this method, a tensor is constructed from matrix data by transforming the data columns to Löwner matrices, which are then stacked in a third order tensor. It has been shown in [46] that the resulting tensor for our dataset can be approximated using a BTD in rank-$(L,L,1)$ terms. The goal is to determine whether two mixtures consist of the same components without explicitly identifying them.

In a first experiment, we start by tensorizing the two available mixtures using Löwnerization along the emission mode. It has been shown in [46] that the resulting tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)} \in \mathbb{C}^{101\times 100\times 61}$ admit an approximate BTD in three rank-$(L,L,1)$ terms with $L=2$. This implies that each of the analytes contributes one rank-$(2,2,1)$ term. Though we know that the mixtures consist of three components, one could extract the number of components from the data as well. Figure 6.11 shows the first 8 singular values of $\mathbf{T}^{(1)}_{10100\times 61}$ and $\mathbf{T}^{(2)}_{10100\times 61}$, from which it immediately follows that the matrices can be well-approximated with 3 components. The principal angles between both matrices are 0.037, 0.064 and 0.328. As expected, these small values indicates that both mixtures consist of the same components. These values differ from zero due to measurement noise and the presence of Rayleigh scatter in the data [15].

[1] Available at `http://www.models.life.ku.dk/amino_acid_fluo`

Table 6.1: The principal angles clearly indicate which mixtures consist of the same components.

Mixtures to compare	Principal angles (rad)	
Mix1 and Mix2	0.0664	1.3032
Mix1 and Mix3	0.0217	0.0674
Mix2 and Mix3	0.1198	1.2892

For the second experiment, we use the excitation-emission matrices of the separate analytes to contruct synthetic mixtures of two analytes as follows:

$$\mathrm{EE}\,(\mathrm{Mix1}) = \mathrm{EE}\,(\mathrm{Trp}) + \mathrm{EE}\,(\mathrm{Tyr})\,,$$

$$\mathrm{EE}\,(\mathrm{Mix2}) = 0.4\ \mathrm{EE}\,(\mathrm{Tyr}) + 0.8\ \mathrm{EE}\,(\mathrm{Phe})\,,$$

$$\mathrm{EE}\,(\mathrm{Mix3}) = 0.8\ \mathrm{EE}\,(\mathrm{Trp}) + 0.4\ \mathrm{EE}\,(\mathrm{Tyr})\,,$$

in which we use EE (x) as shorthand to denote the excitation-emission matrix of x. The excitation-emission matrices of all three mixtures are then tensorized using Löwnerization as in the previous experiment. The principal angles resulting from the comparison of the mixtures are given in Table 6.1. The table shows small principal angles between the tensors associated with mixtures 1 and 3, which indeed consist of the same components. Note that the other comparisons show one small and one large principal angle, which corresponds to the mixtures having one component in common. Though we did not explicitly generalize Theorem 6 to check partial equality of CPDs to BTDs, we expect similar results for the latter. This seems to be confirmed by the data in Table 6.1.

6.7 Conclusions

We have introduced a method to compare underlying factors of tensor decompositions in two modes without explicitly computing the factors. A step-by-step guide was provided detailing how to proceed in practice. Theoretical theorems were given for CPDs, BTDs in rank-$(L, L, 1)$ terms and BTDs in rank-$(L, L, \cdot)$ terms. Additionally, the partial similarity in two modes of CPDs was considered and we provided a result for non-unique CPDs. The method was illustrated using artificial data and two applications were tackled. The experiments have shown that the method is able to efficiently determine whether the factors in two modes are essentially equal.

6.8 Proof of Theorem 4

Because both $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ admit a BTD in rank-$(L, L, 1)$ terms, their unfoldings can be written as

$$\mathbf{T}^{(1)}_{IJ\times K} = [(\mathbf{B}_1 \odot \mathbf{A}_1)\,\mathbf{1}_L, \ldots, (\mathbf{B}_R \odot \mathbf{A}_R)\,\mathbf{1}_L] \cdot \mathbf{C}^{\mathrm{T}},$$

$$\mathbf{T}^{(2)}_{IJ\times K} = [(\mathbf{E}_1 \odot \mathbf{D}_1)\,\mathbf{1}_L, \ldots, (\mathbf{E}_R \odot \mathbf{D}_R)\,\mathbf{1}_L] \cdot \mathbf{F}^{\mathrm{T}},$$

as explained in Section 6.2.2.

We first show that $\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$ is a necessary condition. Assume $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes, i.e., that $\mathbf{D}_r = \mathbf{A}_r\mathbf{G}_r$ and $\mathbf{E}_r = \mathbf{B}_r\left(\mathbf{G}_r^{-1}\right)^{\mathrm{T}}$ for each $r \in \{1, \ldots, R\}$ with $\mathbf{G}_r$ nonsingular matrices. Also, the different terms may be permuted and we assume without loss of generality that any further scaling is captured in the factors $\mathbf{C}$ and $\mathbf{F}$. Because $(\mathbf{B}_r \odot \mathbf{A}_r)\,\mathbf{1}_L$ is simply the vectorized matrix product $\mathbf{A}_r\mathbf{B}_r^{\mathrm{T}} = \mathbf{A}_r\mathbf{G}_r\mathbf{G}_r^{-1}\mathbf{B}_r{}^{\mathrm{T}} = \mathbf{D}_r\mathbf{E}_r^{\mathrm{T}}$, it follows that

$$\mathbf{T}^{(1)}_{IJ\times K} = [(\mathbf{B}_1 \odot \mathbf{A}_1)\,\mathbf{1}_L, \ldots, (\mathbf{B}_R \odot \mathbf{A}_R)\,\mathbf{1}_L]\,\mathbf{C}^{\mathrm{T}},$$

$$\mathbf{T}^{(2)}_{IJ\times K} = [(\mathbf{B}_1 \odot \mathbf{A}_1)\,\mathbf{1}_L, \ldots, (\mathbf{B}_R \odot \mathbf{A}_R)\,\mathbf{1}_L]\,\mathbf{\Pi}\mathbf{F}^{\mathrm{T}},$$

in which the $\mathbf{\Pi}$ accounts for the possible column permutation. Since $\mathbf{C}$ and $\mathbf{F}$ are assumed to have full column rank and $\mathbf{\Pi}$ is nonsingular, it immediately follows that $\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$.

Conversely, assume $\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$ holds to show that this subspace equality is also sufficient. Because $\mathbf{C}$ and $\mathbf{F}$ have full column rank, the subspace equality can be written as

$$[(\mathbf{B}_1 \odot \mathbf{A}_1)\,\mathbf{1}_L, \ldots, (\mathbf{B}_R \odot \mathbf{A}_R)\,\mathbf{1}_L]\left(\mathbf{M}^{(1)}\right)^{\mathrm{T}}$$

$$= [(\mathbf{E}_1 \odot \mathbf{D}_1)\,\mathbf{1}_L, \ldots, (\mathbf{E}_R \odot \mathbf{D}_R)\,\mathbf{1}_L]\left(\mathbf{M}^{(2)}\right)^{\mathrm{T}}, \tag{6.22}$$

with $\mathbf{M}^{(1)}, \mathbf{M}^{(2)} \in \mathbb{C}^{R\times R}$ nonsingular matrices. Both sides of equation (6.22) represent an unfolded BTD in rank-$(L, L, 1)$ terms of the same tensor.

Consequently, we can write

$$\mathcal{G} = \sum_{r=1}^{R} \left(\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}\right) \circ \mathbf{m}_r^{(1)} \tag{6.23}$$

$$= \sum_{r=1}^{R} \left(\mathbf{D}_r \mathbf{E}_r^{\mathrm{T}}\right) \circ \mathbf{m}_r^{(2)} \in \mathbb{C}^{I \times J \times R}.$$

It follows from (6.9a) that the BTD in rank-$(L, L, 1)$ terms in equation (6.23) is essentially unique. Consequently, the factor matrices $\mathbf{A}$ and $\mathbf{D}$, and $\mathbf{B}$ and $\mathbf{E}$ are equal up to the trivial indeterminacies stated in Section 6.2.2.

6.9 Proof of Theorem 5

We first show that $\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$ is a necessary condition. Assume $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ are essentially equal in the first two modes, i.e., that $\mathbf{C}_r = \mathbf{A}_r\mathbf{G}_r$ and $\mathbf{D}_r = \mathbf{B}_r\left(\mathbf{G}_r^{-1}\right)^{\mathrm{T}}$ for each $r \in \{1, \ldots, R\}$ with $\mathbf{G}_r \in \mathbb{C}^{L\times L}$ nonsingular matrices. Also, the different terms may be permuted and we assume without loss of generality that any further scaling is captured in the factors $\mathbf{C}$ and $\mathbf{F}$. Because $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ have full row rank, bases for the column spaces of $\mathbf{T}^{(1)}_{IJ\times K}$ and $\mathbf{T}^{(2)}_{IJ\times K}$ are given by

$$\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{[\mathbf{B}_1 \otimes \mathbf{A}_1, \ldots, \mathbf{B}_R \otimes \mathbf{A}_R]\right\},$$

$$\operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\} = \operatorname{span}\left\{\left[\left(\mathbf{B}_1\left(\mathbf{G}_1^{-1}\right)^{\mathrm{T}}\right) \otimes (\mathbf{A}_1\mathbf{G}_1), \ldots,\right.\right.$$

$$\left.\left.\left(\mathbf{B}_R\left(\mathbf{G}_R^{-1}\right)^{\mathrm{T}}\right) \otimes (\mathbf{A}_R\mathbf{G}_R)\right]\right\}.$$

Using the Kronecker product property $(\mathbf{UV}) \otimes (\mathbf{WX}) = (\mathbf{U} \otimes \mathbf{W})(\mathbf{V} \otimes \mathbf{X})$, the second basis becomes

$$\operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\} = \operatorname{span}\left\{[\mathbf{B}_1 \otimes \mathbf{A}_1, \ldots, \mathbf{B}_R \otimes \mathbf{A}_R]\right\},$$

from which it immediately follows that $\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$.

Conversely, assume $\operatorname{span}\left\{\mathbf{T}^{(1)}_{IJ\times K}\right\} = \operatorname{span}\left\{\mathbf{T}^{(2)}_{IJ\times K}\right\}$ holds to show that this condition is sufficient as well. Because $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ have full row rank, the

subspace equality can be written as

$$[\mathbf{B}_1 \otimes \mathbf{A}_1, \ldots, \mathbf{B}_R \otimes \mathbf{A}_R] \left(\mathbf{M}^{(1)}\right)^{\mathrm{T}}$$

$$= [\mathbf{D}_1 \otimes \mathbf{C}_1, \ldots, \mathbf{D}_R \otimes \mathbf{C}_R] \left(\mathbf{M}^{(2)}\right)^{\mathrm{T}}, \tag{6.24}$$

with $\mathbf{M}^{(1)}, \mathbf{M}^{(2)} \in \mathbb{C}^{RL^2 \times RL^2}$ nonsingular matrices. Both sides of equation (6.24) represent an unfolded BTD in rank-$(L, L, \cdot)$ terms of the same tensor. Consequently, we can write

$$\mathcal{G} = \sum_{r=1}^{R} \mathcal{M}_r^{(1)} \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r, \tag{6.25}$$

$$= \sum_{r=1}^{R} \mathcal{M}_r^{(2)} \bullet_1 \mathbf{C}_r \bullet_2 \mathbf{D}_r \in \mathbb{C}^{I \times J \times RL^2}.$$

Since it was assumed in (6.12a) that this BTD is essentially unique, the factor matrices $\mathbf{A}$ and $\mathbf{D}$, and $\mathbf{B}$ and $\mathbf{E}$ are equal up to the trivial indeterminacies stated in Section 6.2.2.

Chapter 7

Determining whether two datasets cluster similarly without determining the clusters

Abstract — When several datasets of observations in the same space are available, such as customer data in various countries, a key question is whether these datasets cluster similarly and if they can be merged for further analysis. For this reason, assessing whether two related datasets share the same cluster centers is an important problem. This paper tackles this problem using a novel method based on tensor similarity, which allows us to compare the cluster centers of two datasets without explicitly computing these clusters.

Reference — This chapter is a slightly adapted version of [143]. The candidate did part of the research and wrote the article under the guidance of the coauthor.

7.1 Introduction

In data analysis, several applications call for a comparison of the cluster patterns of two datasets. Consider customer profiling based on consumer data from various countries. If the same profiles hold in different countries, the different datasets can be merged before further analysis. If not, each country must be analyzed separately. Other examples can be found in sets of scientific experiments conducted at different points in time and by different researchers (see [8]), or in survey results to see whether the results from a one-hour survey can be replaced by the results of a ten-minute survey.

In this paper, a measure is constructed that compares the cluster centers of different datasets in the same space. A naive way to tackle this problem is to cluster the datasets separately using any algorithm ([161]) and then to compare the cluster centers. However, this approach is sensitive to the cluster algorithm and how it is initialized.

Metrics to compare various clusterings of the same dataset have been treated many times in literature [97, 159, 158]. These approaches can be roughly divided into three groups: counting pairs of elements, checking set overlaps and information-theoretical mutual information [159]. However, these metrics assume the same underlying dataset, which is not the case in the applications considered in this paper.

Methods that compare cluster centers across datasets are scarce in the literature. In [96], a Bayesian approach is considered that determines whether both datasets can emerge from a statistical model with the same model parameters. The downside of this approach is that statistical modeling is needed. Another approach translates the comparison of datasets to a binary classification problem [102]. First, the datasets are merged and equipped with a source indicator. If a binary classifier is unable to reliably predict this source indicator, the datasets are considered different. Note that this approach requires a classifier to be trained and tuned.

The approach presented in this paper is based on tensor analysis but requires only simple linear algebra. Unlike matrix decompositions, tensor decompositions are unique up to trivial indeterminacies under mild conditions [81, 110]. Recent results in tensor similarity show that it is possible to compare the underlying factors of two tensor decompositions without explicitly computing them [145, 53, 33, 144]. This property can be used to assess whether two separate datasets in the same space share the same cluster centers, yielding a conceptually simple yet powerful method.

Notations

Following notation are used throughout this paper: Scalars are denoted by lowercase letters, vectors by bold lowercase letters, matrices by bold uppercase letters, and tensors by uppercase calligraphic letters (e.g., $\mathcal{A}$). Subsets of vectors, matrices and tensors are denoted by indices (e.g. a_{ij} for the entry of the matrix $\mathbf{A}$). Following symbols are used: outer product ($\circ$), Kronecker product ($\otimes$), Khatri–Rao product ($\odot$), expected value ($E[\cdot]$), Kronecker delta ($\delta(\cdot)$), vector of all ones ($\mathbf{1}$), ith canonical basis vector of a space ($\mathbf{e}_i$).

7.2 Problem statement

Consider two datasets, $\mathbf{V} \in \mathbb{C}^{I \times M}$ and $\mathbf{W} \in \mathbb{C}^{I \times N}$, representing a set of observations in the same I-dimensional feature space. The jth column of $\mathbf{V}$ is denoted by $\mathbf{v}(j)$ and represents the jth data point of the dataset (similarly for $\mathbf{w}(j)$ of $\mathbf{W}$). Assume both datasets display R clusters with cluster centers $\mathbf{\Pi} = [\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_R]$ and $\mathbf{\Gamma} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_R]$ respectively. The probability that a randomly chosen data point $\mathbf{v}(j)$ is part of the rth cluster of dataset $\mathbf{V}$ is denoted by ξ_r (and ζ_r for dataset $\mathbf{W}$). The points of both sets can now be written as

$$\mathbf{v}(j) = \boldsymbol{\pi}(j) + \mathbf{p}(j),$$

$$\mathbf{w}(j) = \boldsymbol{\gamma}(j) + \mathbf{q}(j).$$

In this equation $\boldsymbol{\pi}(j), \boldsymbol{\gamma}(j) \in \mathbb{C}^I$ represent functions that map the index of a data point to its corresponding cluster center $\boldsymbol{\pi}_r$ or $\boldsymbol{\gamma}_r$. The functions $\mathbf{p}(j), \mathbf{q}(j) \in \mathbb{C}^I$ represent the deviation of the jth data point from its corresponding cluster center. Assume following conditions hold:

- Deviations from cluster centers are zero-mean for each cluster, i.e. $E[\mathbf{p}(j)] = \mathbf{0}$ and $E[\mathbf{q}(j)] = \mathbf{0}$.
- The cluster distributions are independent across clusters and dimensions, i.e., $p_k(j)$ is independent from $p_l(j)$ for $k \neq l$. This implies that the covariance matrix of $\mathbf{p}(j)$ (and $\mathbf{q}(j)$) is diagonal. For cluster r of dataset $\mathbf{V}$, the diagonal of the covariance matrix is given by $\boldsymbol{\sigma}^{(r,v)}$.
- The matrices containing the cluster centers $\mathbf{\Pi}, \mathbf{\Gamma} \in \mathbb{C}^{I \times R}$ satisfy following conditions. If tensorization based on Kronecker products, as explained in Section 7.4.2, is used then following condition must hold:

$$\mathcal{C}_2(\mathbf{\Pi}) \odot \mathcal{C}_2(\mathbf{\Pi}) \quad \text{has full column rank.}$$

In this conditions, $\mathcal{C}_2(\mathbf{\Pi})$ denotes the 2nd compound matrix of the $I \times R$ matrix $\mathbf{\Pi}$, which is the $\binom{I}{2} \times \binom{R}{2}$ matrix containing the determinants of all 2×2 submatrices of $\mathbf{\Pi}$ [49]. The determinants are arranged such that the submatrix index sets are in lexicographic order.

If tensorization based on third-order moments is used however, which is explained in Section 7.4.3, then the condition above is no longer applicable and instead following condition should hold:

$$k_{\mathbf{\Pi}} \geq \frac{2}{3}R + \frac{2}{3}.$$

In this, $k_{\mathbf{\Pi}}$ denotes the k-rank of $\mathbf{\Pi}$, which is defined as the largest number $k_{\mathbf{\Pi}}$ such that every subset of $k_{\mathbf{\Pi}}$ of the matrix $\mathbf{\Pi}$ is linearly independent [84].

The goal of this chapter is to assess whether the cluster centers are the same for both datasets without explicitly determining the clusters.

7.3 Tensors

7.3.1 Rank and Canonical Polyadic Decomposition

A rank-1 tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ of order N is defined as the outer product of N nonzero vectors $\mathbf{u}^{(n)} \in \mathbb{C}^{I_n}$, i.e., $\mathcal{A} = \mathbf{u}^{(1)} \circ \cdots \circ \mathbf{u}^{(N)}$. The rank of a tensor $\mathcal{T}$ is then defined as the minimum number of rank-1 terms that yield $\mathcal{T}$ in a linear combination. Such a (minimal) linear combination of rank-1 terms is called a canonical polyadic decomposition (CPD). Mathematically, the CPD of a rank-R tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ can be written as

$$\mathcal{T} = \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(N)} = \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right]. \tag{7.1}$$

In which the factor vectors $\mathbf{u}_r^{(n)} \in \mathbb{C}^{I_n}$ are concatenated in factor matrices $\mathbf{U}^{(n)} = \left[\mathbf{u}_1^{(n)}, \ldots, \mathbf{u}_R^{(n)}\right] \in \mathbb{C}^{I_n \times R}$ and the weighting factors in $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R] \in \mathbb{C}^R$ to allow a more concise notation [81].

One of the key properties of tensors are that their decompositions are unique under mild conditions:

Theorem 7. *Consider the polyadic decomposition of* $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ *as defined in* (7.1)*. If*

$$\begin{cases} \mathbf{U}^{(3)} \in \mathbb{C}^{I_3 \times R} \text{ has full column rank,} \\ \mathcal{C}_2\left(\mathbf{U}^{(1)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(2)}\right) \text{ has full column rank,} \end{cases}$$

then the rank of $\mathcal{X}$ *is* R *and the CPD of* $\mathcal{X}$ *is essentially unique [74, 30, 49, 50].*

In this theorem, $\mathcal{C}_2(\mathbf{A})$ denotes the 2nd compound matrix as defined above.

7.3.2 Tensor similarity

Recent research on tensor similarity shows that it is possible to compare the factors of the decompositions of two (different) tensors without explicitly having to compute these decompositions [33, 53, 145, 144]. The Theorem 8 below shows one specific result of [145] that considers similarity in the first two modes of two rank-R CPDs. The theorem indicates that, under specific (mild) conditions, the factors in the first two modes are essentially equal if and only if the unfolded tensors in the first two modes span the same space:

Theorem 8. *Let two tensors* $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}$ *admit a rank-R CPD as follows:*

$$\mathcal{T}^{(1)} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \in \mathbb{C}^{I \times J \times K},$$

$$\mathcal{T}^{(2)} = [\![\mathbf{D}, \mathbf{E}, \mathbf{F}]\!] \in \mathbb{C}^{I \times J \times K}.$$

If

(i) $\mathbf{C}$ *and* $\mathbf{F}$ *have full column rank,*

(ii) $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ *has full column rank,*

then $\mathcal{T}^{(1)}$ *and* $\mathcal{T}^{(2)}$ *are essentially equal in the first two modes if and only if* $span\left\{\mathbf{T}^{(1)}_{IJ \times K}\right\} = span\left\{\mathbf{T}^{(2)}_{IJ \times K}\right\}$.

The matrices $\mathbf{T}^{(1)}_{IJ \times K} \in \mathbb{C}^{IJ \times K}$ and $\mathbf{T}^{(2)}_{IJ \times K} \in \mathbb{C}^{IJ \times K}$ are matricized versions of the tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ obtained by stacking the columns of each frontal matrix slice [81].

An important note for this theorem lies in the exact definition of essential equality. Since the CPD has trivial scaling and permutation ambiguities, essential equality means that the factors in the first two modes are the same up to column scaling and permutation [145].

7.4 Comparing cluster centers

7.4.1 Step-by-step algorithm

The method presented in this paper consists of three main steps: fixing scaling ambiguities, tensorization and comparison of the tensors.

First, the datasets are extended to fix any scaling ambiguities. Because of the scaling and permutation ambiguities of tensor decompositions, tensor similarity methods can only compare factor matrices up to scaling and permutation. This scaling ambiguity can be fixed by adding normalization data to $\mathbf{V}$ and $\mathbf{W}$ and constructing $\widehat{\mathbf{V}} = \begin{bmatrix}\mathbf{V}^{\mathrm{T}}, \mathbf{1}\end{bmatrix}^{\mathrm{T}}$ and $\widehat{\mathbf{W}} = \begin{bmatrix}\mathbf{W}^{\mathrm{T}}, \mathbf{1}\end{bmatrix}^{\mathrm{T}}$. The additional row of ones fixes the scaling ambiguity, which is easy to see as follows: vectors $[2, 2]^{\mathrm{T}}$ and $[3, 3]^{\mathrm{T}}$ are the same up to scaling, whereas vectors $[2, 2, 1]^{\mathrm{T}}$ and $[3, 3, 1]^{\mathrm{T}}$ are not.

In a second step, the extended datasets are tensorized using one of the techniques presented in Sections 7.4.2 and 7.4.3 depending on the assumptions. This yields two tensors $\mathcal{V} \in \mathbb{C}^{I \times I \times X}$ and $\mathcal{W} \in \mathbb{C}^{I \times I \times Y}$ that contain the respective cluster centers in the factor matrices in the first two modes. The exact dimensions X and Y of the tensors may vary depending on the tensorization method that is used.

Finally, the factor matrices of both tensors $\mathcal{V}$ and $\mathcal{W}$ in the first two modes can be compared using tensor similarity. The high-level steps for one specific tensor similarity method from [145] are briefly repeated here for later reference:

1. Verify whether the conditions of Theorem 8 hold, in which the challenge is that the factor matrices are not available since no explicit decomposition is computed. See [145] and [144] for detailed instructions on how to verify these conditions without directly accessing the factor matrices.

2. Determine the number of clusters R. This value can stem from knowledge of the application or can be estimated. If R is smaller than I^2 and X (or Y), a good estimate for R is the number of dominant singular values of the unfolded tensor $\mathbf{V}_{I^2 \times X}$ (or $\mathbf{W}_{I^2 \times Y}$). The underlying rationale for this estimate is that under the given conditions an unfolded tensor that admits a rank-R CPD is a rank-R matrix since the matrix dimensions are not limiting the rank.

3. Find bases for the R-dimensional dominant column spaces of the unfolded tensors $\mathbf{V}_{I^2 \times X}$ and $\mathbf{W}_{I^2 \times Y}$. This can for instance be done by taking the

left singular vectors corresponding to the R largest singular values of that matrix.

4. The final step consists of computing the R principal angles between the computed bases of the unfolded tensors. These principal angles are the minimum angles between both subspaces. Further details on principal angles can be found in [11, 80].

By going through these steps, the similarity of the R cluster centers of $\mathbf{V}$ and $\mathbf{W}$ has been reduced to a set of R principal angles. In brief, all principal angles will be zero if the cluster centers of both datasets are the same since the scaling ambiguity has been fixed [145].

Note that the approach presented in this paper can easily be adapted for the case where $\mathbf{V}$ and $\mathbf{W}$ only share part of their cluster centers. Without going in full detail, the interested reader is referred to [145] for a theorem that expands on this partial similarity.

7.4.2 Tensorization via Kronecker products

To tensorize the given datasets, compute the Khatri–Rao products $\mathbf{V} \odot \mathbf{V} \in \mathbb{C}^{I^2 \times M}$ and $\mathbf{W} \odot \mathbf{W} \in \mathbb{C}^{I^2 \times N}$:

$$\mathbf{V} \odot \mathbf{V} = [\mathbf{v}(1) \otimes \mathbf{v}(1), \ldots, \mathbf{v}(M) \otimes \mathbf{v}(M)]\,,$$

$$\mathbf{W} \odot \mathbf{W} = [\mathbf{w}(1) \otimes \mathbf{w}(1), \ldots, \mathbf{w}(N) \otimes \mathbf{w}(N)]\,.$$

Now consider the jth column of $\mathbf{V} \odot \mathbf{V}$ and assume without loss of generality that the corresponding jth data point is in cluster r. The $(k + I(l-1))$th entry of this vector is then given by

$$\begin{aligned}
&(\mathbf{v}(j) \otimes \mathbf{v}(j))_{k+I(l-1)} \\
&\quad = (\pi_l(j) + p_l(j))\,(\pi_k(j) + p_k(j)) \\
&\quad = \pi_{lr}\pi_{kr} + p_l(j)p_k(j) + \pi_{kr}p_l(j) + \pi_{lr}p_k(j) \qquad (7.3)
\end{aligned}$$

Analogous expressions can be found for entries of $\mathbf{W} \odot \mathbf{W}$.

Generally, all terms in equation (7.3) are nonzero. However, one step in the tensor similarity procedure computes the dominant subspace of the unfolded tensors $\mathbf{V} \odot \mathbf{V}$ and $\mathbf{W} \odot \mathbf{W}$, which has an averaging effect on the entries of both matrices. During this operation, the terms $\pi_{kr}p_l(j)$ and $\pi_{lr}p_k(j)$ in (7.3) will average to zero because of the assumed zero-mean deviations from cluster

centers. This leaves following expressions for the expected value of entries of $\mathbf{V} \odot \mathbf{V}$:

$$E\left[(\mathbf{v}(j) \otimes \mathbf{v}(j))_{k+I(l-1)}\right] = \pi_{lr}\pi_{kr} + \delta(k-l)\left(\sigma_k^{(r,v)}\right)^2 \tag{7.4}$$

For $j \neq k$, the term $p_l(j)p_k(j)$ averages to zero over the data points of the rth cluster because of the independence of both variables and their zero mean. For $j = k$ however, this term becomes $p_k^2(j)$ and will average to $\left(\sigma_k^{(r,v)}\right)^2$, the variance of the rth cluster in the kth dimension. If the first two assumptions in Section 7.2 do not hold, the terms above generally do not cancel out and must be considered further, which deteriorates the solution.

If the cluster variances remain relatively small in comparison to the norms of the cluster centers, the first term of (7.4) is dominant. This in turn implies that the R-dimensional dominant subspace of $\operatorname{span}\{\mathbf{V} \odot \mathbf{V}\}$ can be approximated by $\mathbf{\Pi} \odot \mathbf{\Pi}$ (and similarly for $\operatorname{span}\{\mathbf{W} \odot \mathbf{W}\}$). Mathematically, this can be written as

$$\mathbf{V} \odot \mathbf{V} \approx (\mathbf{\Pi} \odot \mathbf{\Pi})\,\mathbf{K}^{\mathrm{T}}, \tag{7.5}$$

$$\mathbf{W} \odot \mathbf{W} \approx (\mathbf{\Gamma} \odot \mathbf{\Gamma})\,\mathbf{L}^{\mathrm{T}}, \tag{7.6}$$

in which $\mathbf{K} \in \mathbb{C}^{M \times R}$ and $\mathbf{L} \in \mathbb{C}^{N \times R}$ can be interpreted as matrices that select the correct cluster center for each data point and capture the spread around that cluster center. Alternatively, the rows of $\mathbf{V} \odot \mathbf{V}$ and $\mathbf{W} \odot \mathbf{W}$ that are effected by the cluster variances can be omitted. This does not change anything in the computations, but affects the conditions given in Theorem 8 as the corresponding submatrix must be considered there as well. Details of this "fiber sampling" approach and the impact on uniqueness conditions can be found in [120].

The expressions (7.5) and (7.6) are matricized CPDs, which implies that $\mathbf{V} \odot \mathbf{V}$ and $\mathbf{W} \odot \mathbf{W}$ can be interpreted as matricized tensors $\mathcal{V} \in \mathbb{C}^{I \times I \times M}$ and $\mathcal{W} \in \mathbb{C}^{I \times I \times N}$ that approximately admit a CPD. Because the information in the first two modes consist purely of the cluster centers, it suffices to compare just these modes.

Note that the matrix $\mathbf{K}$ in (7.5) has full column rank by construction. Combining this with the third assumption in Section 7.2 that states that $\mathcal{C}_2(\mathbf{\Pi}) \odot \mathcal{C}_2(\mathbf{\Pi})$ must have full column rank gives the CPD uniqueness conditions from Theorem 7. This uniqueness is needed to ensure that there are no other rank-1 terms in the space $\operatorname{span}\{\mathbf{V} \odot \mathbf{V}\}$ apart from the rank-1 terms stemming from the cluster centers $\mathbf{\Pi} \odot \mathbf{\Pi}$. It is this property that allows us to make a statement on the essential equality of underlying cluster centers by only comparing the spaces spanned by two unfolded tensors.

Large scale case

If the clusters contain many data points, meaning $M, N \gg I$, the computation of $\mathbf{V} \odot \mathbf{V}$ and $\mathbf{W} \odot \mathbf{W}$ may become memory-intensive. Here, several approaches that can be used to circumvent this problem are briefly touched. First, only a subset of the available data points can be considered, though this will reduce the accuracy. Second, subspace updating can be used to update the dominant subspace of e.g. $\mathbf{V} \odot \mathbf{V}$ every time a new column is added to avoid constructing the full matrix [48]. A third way to handle large-scale data is to compute the matrices $(\mathbf{V} \odot \mathbf{V})(\mathbf{V} \odot \mathbf{V})^{\mathrm{T}} \in \mathbb{C}^{I^2 \times I^2}$ and $(\mathbf{W} \odot \mathbf{W})(\mathbf{W} \odot \mathbf{W})^{\mathrm{T}} \in \mathbb{C}^{I^2 \times I^2}$ instead of $\mathbf{V} \odot \mathbf{V}$ and $\mathbf{W} \odot \mathbf{W}$, since their column spaces are the same. This is advantageous because the former matrices are smaller and can be constructed without storing all columns of e.g. $\mathbf{V} \odot \mathbf{V}$. The downside of this approach is that the condition number of the problem is squared.

7.4.3 Tensorization via third-order moments

The previous section focused on the usage of second-order information for the tensorization. It is also possible to consider third-order or even higher-order information. Assume an additional assumption is imposed: the clusters are all spherical Gaussians with variances $\sigma^{(1,v)}, \ldots, \sigma^{(R,v)}$. In this case, it was shown that the third-order moment of the vector $\mathbf{v}(t)$ (or similarly $\mathbf{w}(t)$) reduces to a simple expression due to the Gaussianity of the clusters and terms averaging to zero [110, 68, 7]. More specifically, an entry of the third-order moment tensor of $\mathbf{v}(t)$ can be written as:

$$m_{ijk}^{(3)} = \sum_{r=1}^{R} \xi_r \pi_{ir} \pi_{jr} \pi_{kr} + \ldots$$

$$\sum_{r=1}^{R} \xi_r \left(\sigma^{(r,v)}\right)^2 \left(\pi_{ir}\delta(j-k) + \pi_{jr}\delta(i-k) + \pi_{kr}\delta(i-j)\right).$$

The first term in this equation is a decomposition in rank-1 terms, in which each cluster center contributes one term. The second term contains the influence of the cluster distributions for a select number of entries where two indices coincide. This second term can be precomputed by calculating (see [68]):

$$\begin{aligned} \mathbf{d} &= E\left[\mathbf{v}(t)\left(\mathbf{u}^{\mathrm{T}}(\mathbf{v}(t) - E\left[\mathbf{v}(t)\right])\right)^2\right] \\ &= \sum_{r=1}^{R} \xi_r \left(\sigma^{(r,v)}\right)^2 \boldsymbol{\pi}_r, \end{aligned} \tag{7.7}$$

in which $\mathbf{u}$ represents any unit norm eigenvector corresponding to the smallest eigenvalue of the covariance matrix $E\left[(\mathbf{v}(t) - E\left[\mathbf{v}(t)\right])(\mathbf{v}(t) - E\left[\mathbf{v}(t)\right])^{\mathrm{T}}\right]$. Once $\mathbf{d}$ is computed using expression (7.7), tensor $\mathcal{D} \in \mathbb{C}^{I \times I \times I}$ can be constructed:

$$\mathcal{D} = \sum_{i=1}^{I} \left(\mathbf{d} \circ \mathbf{e}_i \circ \mathbf{e}_i + \mathbf{e}_i \circ \mathbf{d} \circ \mathbf{e}_i + \mathbf{e}_i \circ \mathbf{e}_i \circ \mathbf{d}\right).$$

By subtracting this tensor from the third-order moment tensor, it then immediately follows that:

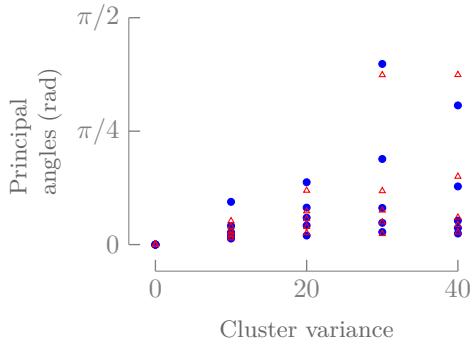
$$\mathcal{M}^{(3)} - \mathcal{D} = \sum_{r=1}^{R} \xi_r \boldsymbol{\pi}^{(r)} \circ \boldsymbol{\pi}^{(r)} \circ \boldsymbol{\pi}^{(r)}.$$

This resulting tensor admitting a CPD can be constructed for both datasets $\mathbf{V}$ and $\mathbf{W}$, and subsequently used to compare the cluster centers in both datasets since each factor matrix contains the cluster centers.

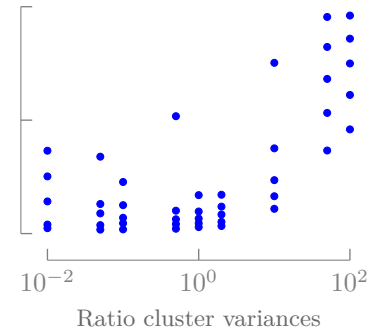
Just as for the tensorization approach using Kronecker products, uniqueness of the tensor decompositions is required to reliably determine whether the underlying cluster centers are the same by only comparing the spaces spanned by two unfolded tensors. In this case however, the conditions from Theorem 7 are too strict since they would require $\mathbf{\Pi}$ to have full column rank. Instead, sufficient uniqueness conditions from [84] are used that indicate that the CPD above is unique if $k_{\mathbf{\Pi}} + k_{\mathbf{\Pi}} + k_{\mathbf{\Pi}} \geq 2R + 2$. This inequality is exactly the same as the third assumption made in the problem statement section 7.2.

7.5 Numerical experiments

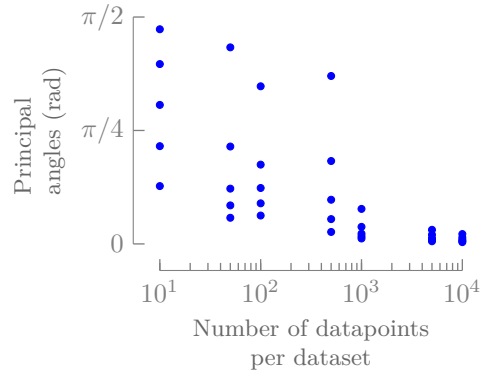
The presented method is illustrated by comparing cluster centers in the context of Gaussian mixture estimation. Consider a mixture of five spherical multivariate Gaussians $\mathcal{N}(\boldsymbol{\pi}_r, \sigma^2\mathbf{I})$. The cluster centers $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_5 \in \mathbb{C}^8$ are chosen randomly from a multivariate normal distribution with zero mean and standard deviation 5. By default, following parameter values are used unless specified otherwise: the cluster variances σ^2 are the same for all clusters and equal to 10. Two datasets are generated from this Gaussian mixture by randomly sampling 1000 data points for each. The probability of a data point belonging to cluster r is the same for all clusters, i.e. $\xi_1 = \ldots = \xi_5 = 0.2$. The cluster centers of these two resulting data sets $\mathbf{V} \in \mathbb{C}^{8 \times 1000}$ and $\mathbf{W} \in \mathbb{C}^{8 \times 1000}$, which are the same by construction, are compared following the method presented in this paper. Figure 7.1 shows the median principal angles over 1000 repetitions of four numerical experiments:



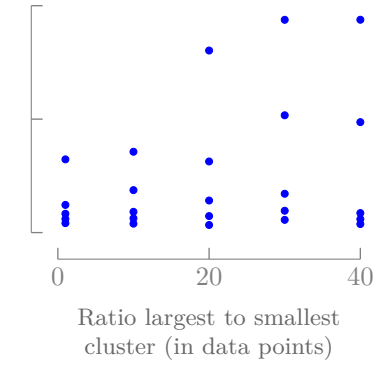
(a) Varying cluster variances with tensorization via Kronecker products (blue dots) and via third-order moments (red triangles).



(b) Varying cluster variances of the datasets, indicated by the ratio of the cluster variance in the second dataset to the first (tensorization via Kronecker products).



(c) Varying the number of data points per dataset (tensorization via Kronecker products).



(d) Varying cluster sizes within datasets, indicated by the ratio of the largest cluster to the smallest (tensorization via third-order moments).

Figure 7.1: Effects of various parameters on the principal angles when the cluster centers of both datasets are the same.

(a) Varying the cluster variances in which all clusters have the same variance. The principal angles are bigger for larger cluster variances since the underlying cluster centers are implicitly estimated less accurately. This experiment shows that both tensorization strategies yield similar results, which is why only one of the methods is depicted in the other experiments.

(b) Varying cluster variances in the second dataset with respect to the first (in which the cluster variances are fixed to 10). The figure shows that even though the principal angles are largely determined by the highest variances, the results slightly worsen for smaller ratios as well. Overall, the principal angles are the smallest when all clusters have approximately the same variance.

(c) Varying the number of datapoints per dataset, which clearly illustrates that more points lead to more accurate comparisons because the underlying estimation of the cluster centers is more accurate.

(d) Varying number of points in the clusters so that the probability of a data point belonging to a specific cluster is no longer uniform. This is expressed as the ratio of the largest to the smallest cluster, where the number of points in each cluster increases linearly and the total number of datapoints per dataset remains fixed at 1000 points. For larger ratios, the performance decreases since it becomes harder to (implicitly) accurately estimate the centers of the smallest clusters.

7.6 Conclusion

This papers introduces a novel method to compare cluster centers of two datasets, which can be used in settings like the comparison of customer profiling in different countries. The method builds on tensor similarity, which allows us to compare the cluster centers without actually determining them. This approach does not require statistical modeling or the development of machine learning algorithms. Two different tensorization options have been presented in the paper, including possible strategies to reduce the memory load of large-scale data. Numerical experiments in a Gaussian mixture estimation context show the effectiveness of the method under various parameter settings.

Chapter 8

Conclusion

This thesis contains new results on tensor-based blind system identification, structured tensor decompositions and tensor similarity. In this final chapter, we give a chapter-by-chapter overview of the main contributions. Next, future possibilities and promising paths are outlined, which can be a starting point for follow-up research.

8.1 Contributions

Chapter 2

- We recognized that tensor-based methods for convolutive ICA usually consist of two steps: a tensorization (and decomposition) step and a deconvolution step. Via numerical experiments, the most appropriate order in which both steps are done is investigated under various conditions.
- Development of a fast deconvolution method. By using only part of the available data, the transformation that is needed to deconvolve the data can be closely approximated. Once this transformation is known, the full dataset can be deconvolved quickly. We showed that this approach is much faster with only a slight loss in accuracy.

Chapter 3

- We derived new generic and deterministic uniqueness conditions for tensor decompositions with at least two block-Hankel structured factors. These bounds exploit the available structure and are much more relaxed than existing results. These new results are applicable on linear FIR blind system identification of systems with i.i.d. inputs, which may even be underdetermined.
- New subspace-based algebraic methods were developed that reduce a banded Hankel-structured tensor decomposition to a third-order unstructured CPD. The algorithms use either the signal or the noise subspace for this. Numerical experiments have shown that these methods are both fast and accurate, which makes them good standalone algebraic algorithms or good initializations for structured optimization-based methods.

Chapter 4

- An overview of the current state of tensor-based methods for BSI with statistically independent inputs was given, together with their main drawbacks in terms of time complexity and accuracy.
- We combined second- and fourth-order statistics of convolutive mixtures in coupled tensor decompositions, following the example from instantaneous ICA. This leads to more accurate results since more information is available. Numerical experiments have shown how the optimal weights in the coupled decomposition are influenced by parameters of the problem.
- We used incomplete tensor decompositions to reduce the construction time of the tensor since we only need to compute a subset of the expensive statistics. This gain in computational speed comes at the cost of a lower accuracy.
- Since decompositions of incomplete tensors are often computed via optimization-based algorithm, a schema is proposed to determine a suitable initialization based on partial information.

Chapter 5

- A new method was presented that efficiently computes a CPD with all block-circulant factors. This methods builds on an existing method in the literature developed for circulant factors. The presented method

transforms the data to the frequency domain and subsequently performs an element-wise devision with a known tensor. This yields a (possibly incomplete) tensor that admits a much simpler decomposition.

Chapter 6

- We presented a method that allows to compare CPD factors of two tensors in two modes, without explicitly computing the factors. For this, a step-by-step guide was provided detailing how to proceed in practice. A theoretical theorem was provided with proof that details under which conditions two CPDs are equal in two modes.
- Just as for CPDs, we developed methods and theorems to compare BTDs in rank-$(L, L, 1)$ terms and in rank-$(L, L, \cdot)$. The theorems are accompanied by proofs.
- We formulated an additional theorem that stipulated when the factors of two CPDs in two modes are partially equal, again with proof. We also show how the common terms can be easily computed and indicate how one could approach the computation of the terms that are different.
- The previous theorems all assume that the CPDs or BTDs are essentially unique. However, we also provided a result that allows to determine when non-unique CPDs share two factors.
- Numerical experiments illustrate what happens when noise is present and when the number of components is estimated incorrectly. The efficiency of the developed methods is additionally illustrated in a synthetic telecommunications application for direction-of-arrival estimation and in a chemometrics example.

Chapter 7

- We show how the comparison of cluster centers of different (but related) datasets can be interpreted as a tensor similarity problem through tensorization.
- Two algorithms are presented to compare cluster centers of two datasets, one based on third-order moments and one on Kronecker-products. Since the approach using Kronecker products may become memory-intensive for large datasets, possible approaches to alleviate the storage requirements of large-scale data in this method are provided.

8.2 Future perspectives

In this section, we highlight possible further research that could build on the work presented in this thesis.

- This thesis focused on BSI of linear FIR systems with statistically independent inputs. We saw that the convolutive structure appeared as block-Toeplitz or bloack-Hankel structured factor matrices. We expect similar structure to appear in related identification problems, such as for infinite impulse response systems and in ARMA systems. The deconvolution techniques and algorithms to decompose structured tensor in this thesis can be used (as inspiration) for work on these related problems.
- The work presented in this thesis focuses on BSI in the time domain, which can be difficult for long filters. An interesting question is whether this information can be combined with frequency-domain methods or time-frequency methods without sacrificing advantages of the respective methods. This is especially of interest when long FIR filters or IIR filters are considered.
- Tensor similarity is a new concept, so there is still a lot of interesting research to be done in this field. This thesis contains theorems to compare factors in two modes for CPDs, BTDs in rank-$(L, L, 1)$ terms and BTDs in rank-$(L, L, \cdot)$. More types of tensor decompositions can be considered in future work, such as BTDs in multilinear rank-(L, M, N) terms, and more than two modes may be considered for third-order tensors. Additionally, one could consider similarity of higher-order tensors (orders larger than three) to see how the results can be further relaxed in these cases.
- Another interesting future perspective is the exploitation of structure. Tensor similarity of structured tensors may further relax the conditions under which tensor similarity may be applied in various applications.
- All algorithms in this thesis are based on mathematical models and are not limited to the examples given throughout this thesis. However, one of the challenges today is to port this theoretical foundation to real-life applications and data. As a future perspective, a few key blind signal separation applications can be chosen where tensors have already showed their potential, such as blind speech separation (preferably in a controlled context such as several people sitting in front of a laptop) and artifact removal in EEGs or MEGs. By further exploiting the available information on signal shape, sparsity and more, the developed models may be further

refined and specialized towards the application at hand, and possible combined with other models to achieve state-of-the-art performance.

Appendix A

Tensor similarity in chemometrics

Abstract — Tensors and their decompositions are widespread in various domains such as signal processing, biomedical data processing and chemometrics. Because of the uniqueness properties of tensor decompositions, tensors have become popular tools to resolve mixtures into their pure components. In some applications however, one wishes to compare two tensors rather than to decompose them. More specifically, one wishes to assess whether two tensors consist of the same rank-1 terms or even block terms in two or more dimensions. Recent advances in tensor analysis have shown that this is possible without having to compute any tensor decompositions, which results in faster and numerically stable algorithms. This chapter aims to introduce this new approach in chemometrics. This is done by explaining the initial insights in tensor similarity, followed by more practical step-by-step approaches which are illustrated with a chemometrics example.

Reference — This chapter is a slightly adapted version of [144] and aims to present the recent advances in tensor similarity to the chemometrics community. A significant part of this chapter concerns the results presented in Chapter 6 (and [145]), where additional details may be found.

A.1 Introduction

Chemometrics has a rich history of extracting information from chemical systems using various statistical or mathematical techniques. Since chemical data is often multi-way by nature, tensors are well-known tools within the community (see [112] and references therein). One popular technique is blind source separation, which tries to resolve spectra of mixtures into the spectra of the pure components. There is a whole body of literature on blind source separation, which is also encountered frequently in domains as biomedical data analysis, digital communications, seismic monitoring and more [28, 36, 71].

Several algorithms for blind source separation exist, including tensor-based techniques, which are the focus of this chapter. Tensor-based blind source separation relies on tensor decompositions to extract the underlying pure components. Again, this approach is strongly embedded within chemometrics community [15], which featured the CP decomposition (also known as PARAFAC) two decades ago.

Recently, new tensor techniques have been developed that go beyond decompositions [33, 145, 53]. It has been shown that the decompositions of two tensors can be compared to each other without having to compute the full decompositions. For instance, one can show that two tensors consist of the same rank-1 terms, possibly differently scaled, without having to compute these rank-1 terms. These new insights and algorithms open up opportunities in all fields where tensor decompositions are used. The goal of this chapter is to introduce these new developments in the field of tensors to a broader audience in chemometrics.

The remainder of this introduction introduces the mathematical notations that are used throughout this chapter. Next, Section A.2 briefly introduces tensors and their main properties. In Section A.3, the relevant tensor decompositions are discussed. Next, tensor similarity is introduced and discussed at length in Section A.4, together with some initial insights. Finally, Section A.5 illustrates some of the concepts with applications in chemometrics.

Notations Scalars are represented by italic lowercase letters (e.g., a), column vectors by bold lowercase letters (e.g., $\mathbf{a}$), matrices by bold uppercase letters (e.g., $\mathbf{A}$) and tensors by uppercase calligraphic letters (e.g., $\mathcal{A}$). Subscripts are used to denote subsets of vectors, matrices and tensors. For instance, the rth column vector of the matrix $\mathbf{A}$ is denoted by $\mathbf{a}_r$. Analogously, t_{ijk} represents the entry with indices (i, j, k) of the tensor $\mathcal{T}$. Slices of a tensor will be denoted using Matlab-style notation. For instance, the kth frontal slice of a tensor $\mathcal{T}$ is

denoted by $\mathbf{T}_{::k}$. Column-wise vectorization of a matrix is denoted by $\mathrm{Vec}\{\mathbf{A}\}$. Consider following matrix for instance:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},$$

then $\mathrm{Vec}\{\mathbf{A}\} = [1, 4, 2, 5, 3, 6]^{\mathrm{T}}$.

Consider the matrix $\mathbf{A} \in \mathbb{C}^{I \times J}$ given by

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

The Kronecker product of $\mathbf{A} \in \mathbb{C}^{I \times J}$ and $\mathbf{B} \in \mathbb{C}^{K \times L}$ is then denoted by $\mathbf{A} \otimes \mathbf{B}$ and given by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \dots & a_{IJ}\mathbf{B} \end{bmatrix} \in \mathbb{C}^{IK \times JL}$$

The Khatri–Rao product of partitioned matrices $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_N]$ and $\mathbf{B} = [\mathbf{B}_1, \dots, \mathbf{B}_N]$ is given by

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{A}_1 \otimes \mathbf{B}_1, \dots, \mathbf{A}_N \otimes \mathbf{B}_N].$$

Unless specified otherwise, matrices in a Khatri–Rao product are assumed to be partitioned columnwise, i.e. $\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \dots, \mathbf{a}_N \otimes \mathbf{b}_N]$. The outer product is denoted by $\circ$ and is defined as follows: consider the outer product of N vectors $\mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \dots \circ \mathbf{u}^{(N)}$, this yields an Nth-order tensor $\mathcal{A}$ of which each entry is given by

$$a_{i_1 i_2 \dots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \dots u^{(N)}_{i_N}.$$

The transpose of a matrix is denoted by $\cdot^{\mathrm{T}}$, the Hermitian transpose by $\cdot^{\mathrm{H}}$ and the column space of a matrix is denoted by $\mathrm{span}\{\mathbf{A}\}$. The tensor-matrix product of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$ and a matrix $\mathbf{M} \in \mathbb{C}^{J_k \times I_k}$ in the kth-mode yields a tensor $\mathcal{B} \in \mathbb{C}^{I_1 \times \dots \times I_{k-1} \times J_k \times I_{k+1} \times \dots \times I_N}$ is denoted by $\mathcal{B} = \mathcal{A} \cdot_k \mathbf{M}$ and is given by

$$b_{i_1, i_2, \dots, i_{k-1}, j_k, i_{k+1}, \dots, i_N} = \sum_{i_k = 1}^{I_k} a_{i_1, i_2, \dots, i_k, \dots, i_N} m_{j, i_k}$$

The intersection of two sets A and B is denoted by $A \cap B$. The notation $\mathbf{a} \perp \mathbf{b}$ is used for two vectors $\mathbf{a} \in C^K$ and $\mathbf{b} \in C^K$ that are perpendicular to each other.

A.2 Tensors and matrices

Tensors are generalizations of vectors and matrices and are sometimes denoted by multi-way arrays. There are two main reasons why tensors have gained popularity in the past decades.

First, they allow a natural representation of multi-way data. For instance, the temperature in a room monitored over time is naturally represented in a four-dimensional tensor, with three spatial dimensions and one temporal dimension. A chemical example of a third-order tensor could be fluorescence emission spectra measured at several excitation wavelengths for several samples. Another example is any kind of spectrum measured chromatographically for several samples [15]. This multi-way representation simplifies further computations and interpretation of the data. If some specific computations require matrices rather than tensors, the tensor entries can be rearranged in a matrix. This process is called matricization or unfolding of a tensor. Unfolding a third-order tensor $\mathcal{X} \in \mathbb{C}^{I \times J \times K}$ to a matrix can be done in various ways, depending on which modes are combined in one matrix dimensions. Here, following definitions are used:

$$\left(\mathbf{X}_{[1,2;3]}\right)_{j+(i-1)J,k} = x_{ijk},$$

$$\left(\mathbf{X}_{[2,3;1]}\right)_{k+(j-1)K,i} = x_{ijk},$$

$$\left(\mathbf{X}_{[1,3;2]}\right)_{k+(i-1)K,j} = x_{ijk}.$$

The second strength of tensors lies in their decompositions, which are unique under mild conditions. This last property does not exist in matrix decompositions, which can be illustrated by considering a 3×3 matrix and its rank-2 decomposition:

$$\begin{bmatrix} 21 & 31 & 6 \\ 27 & 41 & 9 \\ 33 & 51 & 12 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 3 & 2 \\ 5 & 7 & 1 \end{bmatrix}. \tag{A.1}$$

In such a matrix decomposition, it is possible to post- and premultiply the factors with a regular matrix and its inverse without changing the outcome,

e.g.:

$$\begin{bmatrix} 21 & 31 & 6 \\ 27 & 41 & 9 \\ 33 & 51 & 12 \end{bmatrix} = \left(\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} \right) \left(\begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 3 & 2 \\ 5 & 7 & 1 \end{bmatrix} \right),$$

$$= \begin{bmatrix} 5 & 8 \\ 7 & 10 \\ 9 & 12 \end{bmatrix} \begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & -0.5 \end{bmatrix}. \tag{A.2}$$

The example shows two different rank-2 decompositions (A.1) and (A.2) of the same matrix, which immediately illustrates that matrix decompositions are not unique. The only way to obtain unique matrix decompositions is by imposing additional constraints, such as orthogonality in the singular value decomposition. The next section will show that this is in stark contrast with tensor decompositions, which do not need additional constraints and are unique under mild conditions.

A.3 Tensor decompositions

Various tensor decompositions exist, which are used for different objectives such as compression, blind signal separation and classification [81, 110, 25]. This chapter focuses on the multilinear singular value decomposition (MLSVD), the canonical polyadic decomposition (CPD) and several variations of block term decompositions (BTD), which will be used further in the chapter.

A.3.1 Multilinear singular value decomposition

The multilinear singular value decomposition (MLSVD) of a third-order tensor is a multilinear generalization of the well-known singular value decomposition of a matrix [38, 39, 135, 134]. Other terms sometimes used for this decompositions are the higher-order singular value decomposition or the Tucker decomposition. Note that the latter is often used when no additional constraints are imposed on the core tensor or factor matrices of the decomposition.

Consider a third-order tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$. The MLSVD of $\mathcal{A}$ decomposes the tensor as

$$\mathcal{A} = \mathcal{W} \cdot_1 \mathbf{U}^{(1)} \cdot_2 \mathbf{U}^{(2)} \cdot_3 \mathbf{U}^{(3)}, \tag{A.3}$$

in which the factor matrices $\mathbf{U}^{(1)} \in \mathbb{C}^{I_1 \times J_1}$, $\mathbf{U}^{(2)} \in \mathbb{C}^{I_2 \times J_2}$ and $\mathbf{U}^{(3)} \in \mathbb{C}^{I_3 \times J_3}$ are columnwise orthogonal. The core tensor $\mathcal{W} \in \mathbb{C}^{J_1 \times J_2 \times J_3}$ is all-orthogonal

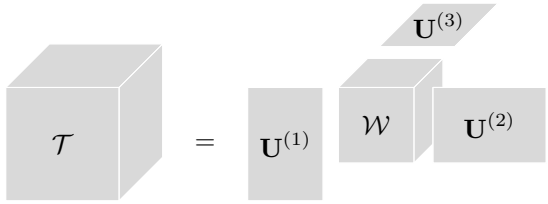


Figure A.1: Graphical illustration of a MLSVD.

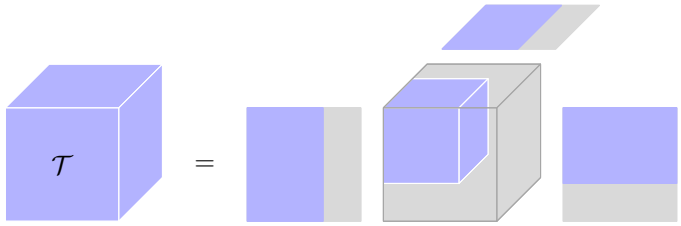


Figure A.2: Illustration of the MLSVD of a tensor with low multilinear rank (assuming the blue core tensor has full multilinear rank). The blue parts of the factor matrices denote the vectors that span the mode-n spaces of $\mathcal{T}$.

and ordered [31]. All-orthogonality means that the different matrix slices in each mode of $\mathcal{W}$ are orthogonal whereas ordered means that the slices in each mode of $\mathcal{W}$ are ordered by descending Frobenius norm. The MLSVD is illustrated graphically in Figure A.1.

The MLSVD is closely linked with the concept of mode-n rank and multilinear rank. The mode-n rank of a tensor $\mathcal{A}$ is the dimension spanned by its mode-n vectors, or equivalently, the rank of its mode-n matrix unfolding $\mathbf{A}_{[k,m;n]}$. The multilinear rank of a third-order tensor $\mathcal{A}$ then equals (I, J, K) if its mode-1, mode-2 and mode-3 rank equal I, J and K, respectively [31].

In a matrix SVD, where $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$, the factor matrices $\mathbf{U}$ and $\mathbf{V}$ contain orthogonal bases for the column space and row space, respectively. The MLSVD generalizes this property to tensors, where the factor matrices $\mathbf{U}^{(n)} \in \mathbb{C}^{I_n \times J_n}$ contain an orthonormal basis for the space spanned by the mode-n vectors of $\mathcal{A}$. More specifically, if $\mathcal{A}$ has multilinear rank (L_1, L_2, L_3) smaller than (J_1, J_2, J_3), then the ordered property of the core tensor $\mathcal{W}$ ensures that the first L_n column vectors of $\mathbf{U}^{(n)}$ form a basis for the space spanned by the mode-n vectors of $\mathcal{A}$, analogous to the SVD in the matrix case. This property is illustrated graphically in Figure A.2.

If the tensor $\widetilde{\mathcal{W}} \in \mathbb{C}^{J_1 \times J_2 \times I_3}$ is defined as $\widetilde{\mathcal{W}} = \mathcal{W} \cdot_3 \mathbf{U}^{(3)}$, then (A.3) can be written as

$$\mathcal{A} = \widetilde{\mathcal{W}} \cdot_1 \mathbf{U}^{(1)} \cdot_2 \mathbf{U}^{(2)}. \tag{A.4}$$

Ignoring the structure of $\widetilde{\mathcal{W}}$, equation (A.4) represents a normalized Tucker-2 decomposition of $\mathcal{A}$ [134].

A.3.2 Canonical polyadic decomposition

The canonical polyadic decomposition or CANDECOMP/PARAFAC is defined as the decomposition of a tensor in a minimal number of rank-1 terms. To understand this definition, the notions of a rank-1 term and a "minimal number" need to be explained. A rank-1 tensor $\mathcal{A}$ of order N is a tensor that can be written as the outer product of N nonzero vectors. Mathematically, this can be written as

$$\mathcal{A} = \mathbf{u}^{(1)} \circ \cdots \circ \mathbf{u}^{(N)},$$

with $\mathbf{u}^{(n)}$ nonzero vectors. Each entry $a_{i_1 i_2 \cdots i_N}$ of the tensor $\mathcal{A}$ is thus given by $a_{i_1 i_2 \cdots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(N)}_{i_N}$.

The rank of a tensor is then defined as the minimum number of rank-1 terms that make up the tensor in a linear combination. Consequently, the CPD of a rank-R tensor decomposes the tensor as a linear combination of R rank-1 tensors. Mathematically, we can write the CPD of a tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times I_N}$ as

$$\mathcal{T} = \sum_{r=1}^{R} \lambda_r \mathbf{u}^{(1)}_r \circ \cdots \circ \mathbf{u}^{(N)}_r, \tag{A.5}$$

in which λ_r is the scaling factor of the rth term. The factor vectors $\mathbf{u}^{(n)}_r \in \mathbb{C}^{I_n}$ are often concatenated in factor matrices $\mathbf{U}^{(n)} = \left[\mathbf{u}^{(n)}_1, \ldots, \mathbf{u}^{(n)}_R\right] \in \mathbb{C}^{I_n \times R}$. Following the notation used in [81], (A.5) can be concisely written as

$$\mathcal{T} = \left[\!\left[\boldsymbol{\lambda}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\right],$$

with $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R] \in \mathbb{C}^R$. A graphical illustration of a CPD is given in Figure A.3.

One important property of the CPD is its essential uniqueness under mild conditions. Essential uniqueness denotes that the decomposition is unique up to two trivial indeterminacies. First, the different rank-1 terms can be arbitrarily permuted. Second, each vector within a rank-1 term can be scaled as long as the other vectors of the rank-1 term are counterscaled. The mathematical

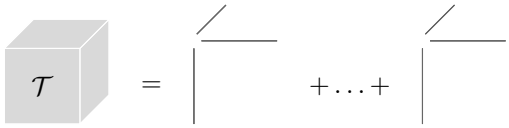


Figure A.3: Graphical illustration of a CPD.

conditions ensuring this essential uniqueness for CPDs have been extensively studied in the literature, see e.g. [52, 50, 110, 84]. Various algorithms to compute (constrained) CPDs have been developed as well, see e.g. [110, 113, 151] and references therein.

Fluorescence spectrography example: Consider five samples containing different amounts of the same three unknown pure analytes, which we wish to retrieve. Denote the excitation-emission matrix of the rth sample as $\mathbf{E}_r \in \mathbb{C}^{K\times N}$, with K excitation wavelengths and N wavelengths at which the emitted light is measured. Let us now stack these five matrices to obtain a tensor $\mathcal{E} \in \mathbb{C}^{K\times N\times 5}$. In [86, 87, 15], the theoretical multilinearity of this tensor was shown for sample with low absorbance and without interactions between chromophores. The paper showed that this tensor admits a rank-R CPD with R the number of pure analytes. The first factor matrix of this CPD contains excitation information, the second one emission information and the third contains the concentrations of the pure analytes in each mixture. To apply this theory to the data at hand, decompose the tensor $\mathcal{E}$ in three rank-1 terms as

$$\mathcal{E} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{r=1}^{3} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r,$$

with $\mathbf{A} \in \mathbb{C}^{K\times 3}, \mathbf{B} \in \mathbb{C}^{N\times 3}$ and $\mathbf{C} \in \mathbb{C}^{5\times 3}$. Any scaling of the different terms was incorporated in the factor matrix $\mathbf{C}$. Because the CPD is essentially unique under mild conditions, which we assume are fulfilled in this case, we know that the factor matrices are unique up to scaling and permutation. Consequently, we can reconstruct the excitation-emission matrices of the pure analytes by computing the outer product of the columns of the first two factor matrices ($\mathbf{A}$ and $\mathbf{B}$) as follows:

$$\text{Excitation-emission matrix of Analyte 1} = \mathbf{a}_1 \circ \mathbf{b}_1 \coloneqq \mathbf{a}_1 {\mathbf{b}_1}^{\mathrm{T}},$$

$$\text{Excitation-emission matrix of Analyte 2} = \mathbf{a}_2 \circ \mathbf{b}_2 \coloneqq \mathbf{a}_2 {\mathbf{b}_2}^{\mathrm{T}},$$

$$\text{Excitation-emission matrix of Analyte 3} = \mathbf{a}_3 \circ \mathbf{b}_3 \coloneqq \mathbf{a}_3 {\mathbf{b}_3}^{\mathrm{T}}.$$

By comparing these reconstructed excitation-emission matrices with the known spectra of pure analytes, the components of the mixtures can be identified. Note that due to the scaling ambiguity, the excitation-emission matrices could be scaled differently, so normalization of the matrices may be required before comparing them with known spectra of pure analytes.

A.3.3 Block term decompositions

In several applications, the rank-1 assumption on each term in a CPD may be too strict to model reality. Because of this, a series of more general decompositions was developed in [31], which are denoted by block-term decompositions (BTD). In this section, a few relevant block-term decompositions are briefly introduced and discussed.

A.3.4 Block term decomposition in rank-$(L, L, 1)$ terms

A BTD in multilinear rank-$(L, L, 1)$ terms is a generalization of a CPD that allows low-rank contributions in the first two modes of each term [31]. This allows for more flexible modeling in applications where the rank-1 assumption is too strict. A graphical representation of a BTD in multilinear rank-$(L, L, 1)$ terms is given in Figure A.4. Mathematically, the BTD of a third-order tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ in rank-$(L, L, 1)$ terms can be written as

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

in which $\mathbf{A}_r \in \mathbb{C}^{I \times L}$ and $\mathbf{B}_r \in \mathbb{C}^{J \times L}$ have full column rank and $\mathbf{c}_r \in \mathbb{C}^{K}$. Essential uniqueness conditions for this decomposition can be found in [31, 32, 117, 54]. Just as in the CPD case, the factor matrices are unique under mild conditions up to trivial indeterminacies. More specifically, the terms can again be arbitrarily permuted, but the scaling indeterminacy is slightly more complex than before. Within each term, one can postmultiply $\mathbf{A}_r$ by any nonsingular $\mathbf{G}_r \in \mathbb{C}^{L \times L}$ as long as $\mathbf{B}_r$ is postmultiplied by $\left(\mathbf{G}_r^{-1}\right)^{\mathrm{T}}$. Moreover, the factors within each multilinear rank-$(L, L, 1)$ term can be scaled and counterscaled as long as their product remains the same. Various algorithms for the computation of this type of decomposition have been developed and can be found in [113, 41, 16, 54, 151].

Fluorescence spectrography example: Let us return to the same problem statement as in the CPD example, but now only one sample is available



Figure A.4: Graphical illustration of a BTD in multilinear rank-$(L, L, 1)$ terms.

containing different amounts of three unknown pure analytes, which we want to retrieve. The introduction showed that matrix decompositions are not unique, so tensorization is used to transform the excitation-emission matrix $\mathbf{E}$ of the sample to a tensor [44, 45]. Many tensorization techniques exist, and choosing the right one is an art on its own. In the context of blind source separation, the suitable tensorization technique depends on the assumptions on the source signals [45]. For example, in ICA the sources are assumed to be statistically independent. It has been shown that this independence assumption can be exploited by computing higher-order statistics of the output signals. If the sources are indeed independent, this yields a tensor that admits a CPD of which the factor matrices contain the mixing information [28]. Analogously, if the sources are exponential polynomials, it has been shown that Hankelization leads to a tensor that allows unmixing [32]. In this fluorescence spectroscopy example, the sources are the emission spectra of the pure analytes. These sources are not statistically independent, but they can be approximated using rational functions, which in turn can be exploited using a tensorization technique called Löwnerization [46]. A Löwner matrix $\mathbf{L}$ has entries

$$l_{i,j} = \frac{f(x_i) - f(y_i)}{x_i - y_i},$$

in which $f(t)$ is a function sampled on N different points t_i and this point set $T = t_1, t_2, \ldots, t_N$ is partitioned in two point sets $X = x_1, x_2, \ldots, x_\alpha$ and $Y = y_1, y_2, \ldots, y_{N-\alpha}$. Note that if N is even and $\alpha = N/2$, then $\mathbf{L}$ is square. The parameter α is typically chosen to be $\lceil N/2 \rceil$. Further background information and details on Löwner matrices can be found in [46] and references therein. To apply Löwnerization to the chemometrics example, each data row along the emission mode is transformed to a Löwner matrix as detailed in [46]. By stacking these matrices, a tensor $\mathcal{E}$ is obtained. It has been shown in [46] that this tensor admits a BTD in three rank-$(L, L, 1)$-terms, with L depending on the spectra of the pure analytes. For a mixture of Tyrosine, Phenylalanine and Tryptophan, a value of $L = 2$ was used in [46]. The number of components R (or more precisely, a lower bound on R) can be estimated as the number of dominant singular values of $\mathbf{E}_{[1,2;3]}$. In this example, $R = 3$ by construction. Similarly, (a lower bound on) the value of RL can be estimated as the number of dominant singular values of $\mathbf{E}_{[2,3;1]}$. Indeed, in this example $\mathbf{E}_{[2,3;1]}$ has 6 dominant singular values. Dividing by R then yields an estimate $L = 2$.

Computing a BTD in three rank-$(2, 2, 1)$-terms of $\mathcal{E}$ yields one block-term per pure analyte:

$$\mathcal{E} = \sum_{r=1}^{3} \left(\widehat{\mathbf{A}}_r \widehat{\mathbf{B}}_r^{\mathrm{T}} \right) \circ \widehat{\mathbf{c}}_r \in \mathbb{C}^{101 \times 100 \times 61},$$

in which the factor matrix $\widehat{\mathbf{C}}$ contains the estimated excitation spectra. If this decomposition is unique up to trivial indeterminacies, the mixture is effectively separated. Several approaches to retrieve the original emission spectra are given in [46]. The most straightforward technique is to invert the factor matrix $\widehat{\mathbf{C}}$ and compute its matrix product with the excitation-emission matrix $\mathbf{E}$:

$$\widehat{\mathbf{S}} = \widehat{\mathbf{C}}^{\dagger} \mathbf{E}$$

The matrix $\widehat{\mathbf{S}}$ then contains the emission spectra of the three underlying pure analytes. Note that this approach is only valid for overdetermined mixtures with a full column rank mixing matrix, see [46] for further details and alternatives.

A.3.5 Block term decomposition in rank-$(L, L, \cdot)$ terms

A BTD in multilinear rank-$(L, L, \cdot)$ is similar to the previous block-term decomposition in the first two modes but has no factor matrix in the third mode. Mathematically, the BTD of a third-order tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ in rank-$(L, L, \cdot)$ terms can be written as

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{C}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r,$$

in which $\mathbf{A}_r \in \mathbb{C}^{I \times L}$ and $\mathbf{B}_r \in \mathbb{C}^{J \times L}$ have full column rank and $\mathcal{C}_r \in \mathbb{C}^{L \times L \times K}$ has mode-1 rank and mode-2 rank equal to L [31]. A graphical representation of a BTD in multilinear rank-$(L, L, \cdot)$ terms is given in Figure A.5. Essential uniqueness conditions for this decomposition can be found in [31, 55]. The trivial ambiguities in the decomposition are the following: the different terms can be arbitrarily permuted and within each term, one can postmultiply $\mathbf{A}_r$ by any nonsingular $\mathbf{G}_r \in \mathbb{C}^{L \times L}$ and $\mathbf{B}_r$ by any nonsingular $\mathbf{H}_r \in \mathbb{C}^{L \times L}$ as long as the core tensor $\mathcal{C}_r$ is replaced by $\mathcal{C}_r \bullet_1 (\mathbf{G}_r)^{-1} \bullet_2 (\mathbf{H}_r)^{-1}$. Algorithms for the computation of this type of decomposition can be found in [41, 55] and toolboxes that implement these algorithms are available as well [156, 157].

A.3.6 Block term decomposition in rank-(L, M, N) terms

A block term decomposition (BTD) in multilinear rank-(L, M, N) is the most general block term decomposition and writes a tensor as a sum of core tensors,

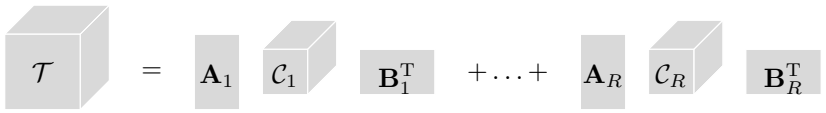


Figure A.5: Graphical illustration of a BTD in multilinear rank-$(L, L, \cdot)$ terms.

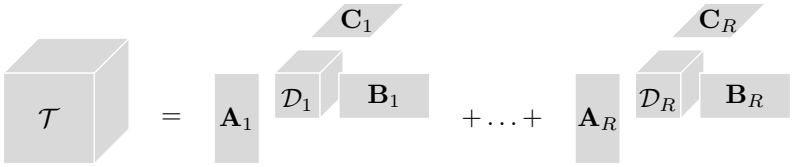


Figure A.6: Graphical illustration of a BTD in multilinear rank-(L, M, N) terms.

each multiplied by a factor matrix in every mode. Mathematically, the BTD of a third-order tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ in rank-(L, M, N) terms can be written as

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

in which $\mathcal{D}_r \in \mathbb{C}^{L \times M \times N}$ are full rank-(L, M, N) and $\mathbf{A}_r \in \mathbb{C}^{I \times L}, \mathbf{B}_r \in \mathbb{C}^{J \times M}$ and $\mathbf{C}_r \in \mathbb{C}^{K \times N}$ have full column rank [31]. A tensor $\mathcal{D}_r \in \mathbb{C}^{L \times M \times N}$ has full rank-(L, M, N) if its matrix unfoldings in the first, second and third mode have rank L, M and N, respectively. A graphical representation of a BTD in multilinear rank-(L, M, N) terms is given in Figure A.6. Essential uniqueness conditions for this decomposition can be found in [31]. The trivial ambiguities in the decomposition are the following: the different terms can be arbitrarily permuted and within each term, one can postmultiply $\mathbf{A}_r$ by any nonsingular $\mathbf{F}_r \in \mathbb{C}^{L \times L}$, $\mathbf{B}_r$ by any nonsingular $\mathbf{G}_r \in \mathbb{C}^{M \times M}$ and $\mathbf{C}_r$ by any nonsingular $\mathbf{H}_r \in \mathbb{C}^{N \times N}$ as long as $\mathcal{D}_r$ is replaced by $\mathcal{D}_r \bullet_1 (\mathbf{F}_r)^{-1} \bullet_2 (\mathbf{G}_r)^{-1} \bullet_3 (\mathbf{H}_r)^{-1}$. Algorithms for the computation of this type of decomposition can be found in [41].

A.4 Tensor similarity

Rather than computing the underlying factors of a tensor, one may wish to compare the factors of two tensors. This can be done by decomposing both tensors and subsequently comparing their factor matrices. However, recent advances in tensor analysis have shown that it is possible to retrieve information

on the decompositions of tensors without explicitly computing them [33, 53, 145]. This approach is computationally more efficient and is numerically more stable [145].

This chapter explains and illustrates how information on tensor decompositions and their interrelations can be obtained without computing the actual tensor decompositions. Section A.4.1 first defines the problem statement in more detail, after which the approach and underlying theory for several cases is treated in sections A.4.2 to A.4.3.

A.4.1 Problem statement

The concept of tensor similarity was introduced in [33]. First note that if two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{N \times N}$ are available with $\mathbf{A}$ having full column rank, knowledge that $\mathbf{B}$ has the same column and row spaces as $\mathbf{A}$ does not provide any information on the structure of $\mathbf{B}$ apart from the obvious result that $\mathbf{B}$ must also have full column rank. However, the situation is different when higher-order tensors are involved, where insights can be derived from the column and the row spaces of various matrix unfoldings.

Consider two third-order tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ for which it holds that

$$\operatorname{span}\left\{\mathbf{A}_{[2,3;1]}\right\} = \operatorname{span}\left\{\mathbf{B}_{[2,3;1]}\right\} \quad \text{and} \quad \operatorname{span}\left\{\mathbf{A}_{[1,3;2]}\right\} = \operatorname{span}\left\{\mathbf{B}_{[1,3;2]}\right\} \tag{A.6}$$

and possibly also that

$$\operatorname{span}\left\{\mathbf{A}_{[1,2;3]}\right\} = \operatorname{span}\left\{\mathbf{B}_{[1,2;3]}\right\}. \tag{A.7}$$

This chapter aims to explain what can be learned about $\mathcal{A}$ and $\mathcal{B}$ and their decompositions from (A.6) and (A.7).

Section A.4.2 shows how to determine which decompositions $\mathcal{A}$ and $\mathcal{B}$ admit and how these decompositions are related based solely on equations (A.6) and (A.7). In particular, we examine if we can infer "essential equality" of the decompositions of $\mathcal{A}$ and $\mathcal{B}$ from (A.6) and (A.7). Essential equality means that both decompositions have the same factor matrices up to the trivial indeterminacies associated with the overall decomposition. Note that if the decomposition is not unique, essential equality means that there exist a pair of decompositions of $\mathcal{A}$ and $\mathcal{B}$ that are essentially equal. Essential equality in two modes means that only the factor matrices in two modes are equal up to trivial indeterminacies, as illustrated graphically for CPDs in Figure A.7.

In section A.4.3 prior information of the type of tensor decomposition is taken into account as well. If it is known that $\mathcal{A}$ and $\mathcal{B}$ admit CPDs or BTDs in

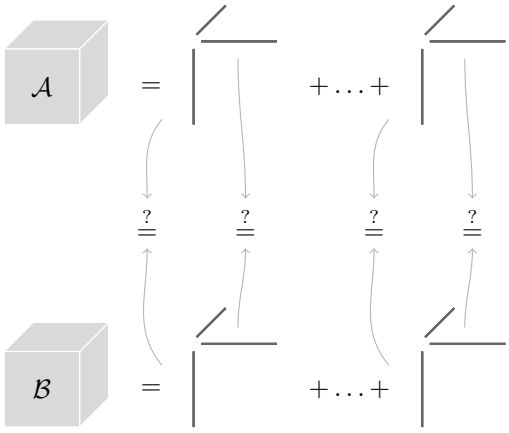


Figure A.7: Graphical representation of comparing the factor matrices of $\mathcal{A}$ and $\mathcal{B}$ to investigate essential equality in the first two modes for the case in which both tensors admit a CPD. (Adopted with changes from [145])

multilinear rank-$(L, L, 1)$ terms, and the number of terms in known, the problem can be further simplified.

Note that the spaces spanned by the matrix unfoldings above are very different from the spaces spanned by the mode-n fibers. The spaces in (A.6) and (A.7) are spanned by the combination of two tensor modes. This is a key element in our analysis of similarity.

A.4.2 Tensor similarity without assumptions on decompositions

Assume two tensors have coinciding spaces spanned by their matrix unfoldings in two dimensions as in equation (A.6), possibly even in three dimensions if (A.7) holds as well. In [33], it is shown how tensor structures and interrelations can be derived from this information in two cases for third-order tensors.

The first case considers two third-order tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ which both have mode-3 rank at least 2 and which both can be simultaneously compressed to tensors of dimensions $(N \times N \times I_3)$ such that the compressed version of $\mathcal{A}$ has full mode-1 and mode-2 rank. Additionally, assume that

$$\text{span}\left\{\mathbf{A}_{[2,3;1]}\right\} = \text{span}\left\{\mathbf{B}_{[2,3;1]}\right\} \quad \text{and} \quad \text{span}\left\{\mathbf{A}_{[1,3;2]}\right\} = \text{span}\left\{\mathbf{B}_{[1,3;2]}\right\} \tag{A.8}$$

It follows from (A.8) that after compression, there must exist exactly two nonsingular matrices $\mathbf{M}_1 \in \mathbb{C}^{N\times N}$ and $\mathbf{M}_2 \in \mathbb{C}^{N\times N}$ such that

$$\mathcal{A} \cdot_1 \mathbf{M}_1 = \mathcal{B} \quad \text{and} \quad \mathcal{A} \cdot_2 \mathbf{M}_2 = \mathcal{B}.$$

Because of the interrelations between these first two modes, it is shown in [33] that the structures of $\mathcal{A}$ and $\mathcal{B}$ are strongly related and depend on the eigenvalues of $\mathbf{M}_2$ (or $\mathbf{M}_1$, which has the same eigenvalues [53]) and their multiplicities. More specifically, following scenarios are identified in [33]:

- If all eigenvalues of $\mathbf{M}_2$ are identical and equal to α and $\mathbf{M}_2$ can be diagonalized, then $\mathcal{A}$ and $\mathcal{B}$ are scaled versions of each other with scaling factor α. It thus holds that $\mathcal{B} = \alpha\mathcal{A}$.
- If all eigenvalues of $\mathbf{M}_2$ are distinct and equal to $\alpha_1, \alpha_2, \ldots, \alpha_N$, then $\mathcal{A}$ and $\mathcal{B}$ consist of the same rank-1 terms, but scaled by the eigenvalues of $\mathbf{M}_2$. Both tensors can be written as

$$\mathcal{A} = \sum_{n=1}^{N} \mathbf{x}_n \circ \mathbf{y}_n \circ \mathbf{z}_n,$$

$$\mathcal{B} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n \circ \mathbf{y}_n \circ \mathbf{z}_n.$$

- If $\mathbf{M}_2$ can be diagonalized and has R eigenvalues α_r with multiplicity L_r, then $\mathcal{A}$ and $\mathcal{B}$ consist of the same rank-$(L_r, L_r, \cdot)$ terms, again scaled by the eigenvalues of $\mathbf{M}_2$. In this case, both tensors can be written as

$$\mathcal{A} = \sum_{r=1}^{R} \mathcal{Z}_r \cdot_1 \mathbf{X}_r \cdot_2 \mathbf{Y}_r,$$

$$\mathcal{B} = \sum_{r=1}^{R} \alpha_r \mathcal{Z}_r \cdot_1 \mathbf{X}_r \cdot_2 \mathbf{Y}_r.$$

- Assume $\mathbf{M}_2$ has R eigenvalues α_r with multiplicity L_r. For $\alpha_2, \ldots, \alpha_R$ the geometric multiplicity is equal to the algebraic multiplicity and for α_1 the geometric multiplicity is equal to 1. In this case, $\mathcal{A}$ and $\mathcal{B}$ consist of the same (possibly differently scaled) rank-$(L_r, L_r, \cdot)$ terms, apart from one block term of which the core is multiplied by a Jordan cell, which has

the form

$$\begin{bmatrix} \alpha_1 & 1 & 0 & \cdots & 0 \\ 0 & \alpha_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \alpha_1 & 1 \\ 0 & 0 & 0 & 0 & \alpha_1 \end{bmatrix} \in \mathbb{C}^{L_1 \times L_1}.$$

If this Jordan cell is denoted by $\mathbf{J}(\alpha_1)$, then both tensors can be written as

$$\mathcal{A} = \sum_{r=1}^{R} \mathcal{Z}_r \bullet_1 \mathbf{X}_r \bullet_2 \mathbf{Y}_r,$$

$$\mathcal{B} = \mathcal{Z}_1 \bullet_1 \mathbf{X}_1 \bullet_2 \left(\mathbf{Y}_1 \mathbf{J}^{\mathrm{T}}(\alpha_1)\right) + \sum_{r=2}^{R} \alpha_r \mathcal{Z}_r \bullet_1 \mathbf{X}_r \bullet_2 \mathbf{Y}_r.$$

- More scenarios are possible based on the eigenvalues of $\mathbf{M}_2$ and their multiplicities.

These scenarios show that if (A.8) holds and $\mathcal{A}$ has full mode-1 and mode-2 rank after compression, the user can first compute $\mathbf{M}_2$ and subsequently find its eigenvalues to determine the structure and interrelations of two tensors $\mathcal{A}$ and $\mathcal{B}$.

In a second case, the matrix unfoldings of both tensors should span the same space for all three dimensions. Mathematically, two third-order tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ are considered such that they can both be simultaneously compressed to tensors of size $(N \times N \times N)$ with the compressed version of $\mathcal{A}$ having full multilinear rank. If following conditions hold:

$$\begin{aligned} &\operatorname{span}\left\{\mathbf{A}_{[2,3;1]}\right\} = \operatorname{span}\left\{\mathbf{B}_{[2,3;1]}\right\}, \quad \operatorname{span}\left\{\mathbf{A}_{[1,3;2]}\right\} = \operatorname{span}\left\{\mathbf{B}_{[1,3;2]}\right\} \\ &\qquad \text{and} \quad \operatorname{span}\left\{\mathbf{A}_{[1,2;3]}\right\} = \operatorname{span}\left\{\mathbf{B}_{[1,2;3]}\right\}, \end{aligned} \tag{A.9}$$

it follows as before that after compression, there must exist nonsingular matrices $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3 \in \mathbb{C}^{N \times N}$ such that

$$\mathcal{A} \bullet_1 \mathbf{M}_1 = \mathcal{B}, \qquad \mathcal{A} \bullet_2 \mathbf{M}_2 = \mathcal{B} \quad \text{and} \quad \mathcal{A} \bullet_3 \mathbf{M}_3 = \mathcal{B}.$$

Similarly as in the first case, the interrelations between all three modes link the structures of both tensors to each other [33]. The tensor structures depend on the eigenvalues of $\mathbf{M}_2$ (or $\mathbf{M}_1$ or $\mathbf{M}_3$) and their multiplicities. Just as in the previous case, this leads to several scenarios:

- If all eigenvalues of $\mathbf{M}_2$ are equal to α and $\mathbf{M}_2$ can be diagonalized, then $\mathcal{A}$ and $\mathcal{B}$ are scaled versions of each other with scaling factor α. It thus holds that $\mathcal{B} = \alpha\mathcal{A}$.
- If all eigenvalues of $\mathbf{M}_2$ are distinct and equal to $\alpha_1, \alpha_2, \ldots, \alpha_N$, then $\mathcal{A}$ and $\mathcal{B}$ consist of the same rank-1 terms, but scaled by the eigenvalues of $\mathbf{M}_2$. Both tensors can be written as

$$\mathcal{A} = \sum_{n=1}^{N} \mathbf{x}_n \circ \mathbf{y}_n \circ \mathbf{z}_n,$$

$$\mathcal{B} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n \circ \mathbf{y}_n \circ \mathbf{z}_n.$$

- If $\mathbf{M}_2$ can be diagonalized and has R eigenvalues α_r with multiplicity L_r, then $\mathcal{A}$ and $\mathcal{B}$ consist of the same rank-(L_r, L_r, L_r) terms, scaled by the eigenvalues of $\mathbf{M}_2$. In this case, both tensors can be written as

$$\mathcal{A} = \sum_{r=1}^{R} \mathcal{W}_r \bullet_1 \mathbf{X}_r \bullet_2 \mathbf{Y}_r \bullet_3 \mathbf{Z}_r,$$

$$\mathcal{B} = \sum_{r=1}^{R} \alpha_r \mathcal{W}_r \bullet_1 \mathbf{X}_r \bullet_2 \mathbf{Y}_r \bullet_3 \mathbf{Z}_r.$$

- Assume $\mathbf{M}_2$ has R eigenvalues α_r with multiplicity L_r. For $\alpha_2, \ldots, \alpha_R$ the geometric multiplicity is equal to the algebraic multiplicity and for α_1 the geometric multiplicity is equal to 1. In this case, $\mathcal{A}$ and $\mathcal{B}$ consist of the same (possibly differently scaled) rank-(L_r, L_r, L_r) terms, apart from one block term of which the core is multiplied by a Jordan cell $\mathbf{J}(\alpha_1)$ associated with α_1. Both tensors can then be written as

$$\mathcal{A} = \sum_{r=1}^{R} \mathcal{W}_r \bullet_1 \mathbf{X}_r \bullet_2 \mathbf{Y}_r \bullet_3 \mathbf{Z}_r,$$

$$\mathcal{B} = \mathcal{W}_1 \bullet_1 \mathbf{X}_1 \bullet_2 \left(\mathbf{Y}_1 \mathbf{J}^{\mathrm{T}}(\alpha_1)\right) \bullet_3 \mathbf{Z}_1 + \sum_{r=2}^{R} \alpha_r \mathcal{W}_r \bullet_1 \mathbf{X}_r \bullet_2 \mathbf{Y}_r \bullet_3 \mathbf{Z}_r.$$

- Depending on the eigenvalues of $\mathbf{M}_2$ (or $\mathbf{M}_1$ or $\mathbf{M}_3$) and their multiplicities, other scenarios are possible.

Both cases above show that coinciding spaces spanned by tensor unfoldings indicate that the decompositions of those tensors are related. The exact structure of the tensors can to a large extent be derived from the eigenvalues of the matrices $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$. In practice, following steps may be followed for both cases to determine how two third-order tensors are related:

1. **Verify the rank conditions:** The first step is to verify whether the (multilinear) rank conditions on $\mathcal{A}$ hold. This step is split up according to the number of modes in which the spaces spanned by the tensor unfoldings should coincide:

   - **Coinciding spaces of tensor unfoldings in two modes:** The rank conditions on $\mathcal{A}$ can be checked by computing its multilinear rank and verifying whether the mode-1 and mode-2 rank are equal and the mode-3 rank is at least two.

     If the mode-3 rank of $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ is only 1, this implies that all frontal slices of $\mathcal{A}$ are scaled versions of each other. Since there is no additional structure in the third mode, this situation is equivalent to the matrix case. The only conclusion one can draw is that the mode-1 and mode-2 ranks of $\mathcal{B}$ must be the same as those of $\mathcal{A}$.

     If $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ has mode-1 and mode-2 rank equal to $N \leq I_1, I_2$, then it is possible to compress both tensors such that their compressed versions do have full mode-1 and mode-2 rank. This can be done as follows: first construct an orthonormal basis for the row space of $\mathbf{A}_{[2,3;1]}$. This can be done by computing the singular value decomposition of $\mathbf{A}_{[2,3;1]}^{\mathrm{T}}$ and storing the left singular vectors corresponding to the N dominant singular values as columns in a matrix $\mathbf{U}_1 \in \mathbb{C}^{I_1 \times N}$. Similarly, an orthonormal basis for the row space of $\mathbf{A}_{[1,3;2]}$ can be constructed and stored in $\mathbf{U}_2 \in \mathbb{C}^{I_2 \times N}$. Next, both tensors can be transformed as

     $$\widetilde{\mathcal{A}} = \mathcal{A} \cdot_1 \mathbf{U}_1^{\mathrm{H}} \cdot_2 \mathbf{U}_2^{\mathrm{H}}, \tag{A.10}$$

     $$\widetilde{\mathcal{B}} = \mathcal{B} \cdot_1 \mathbf{U}_1^{\mathrm{H}} \cdot_2 \mathbf{U}_2^{\mathrm{H}}. \tag{A.11}$$

     Because both tensors are transformed using the same matrices and (A.8) is assumed to hold, their interrelations are not altered [53] and it still holds that

     $$\operatorname{span}\left\{\widetilde{\mathbf{A}}_{[2,3;1]}\right\} = \operatorname{span}\left\{\widetilde{\mathbf{B}}_{[2,3;1]}\right\} \quad \text{and}$$

     $$\operatorname{span}\left\{\widetilde{\mathbf{A}}_{[1,3;2]}\right\} = \operatorname{span}\left\{\widetilde{\mathbf{B}}_{[1,3;2]}\right\}.$$

Moreover, the matrix unfoldings of $\widetilde{\mathcal{A}}$ have full column rank by construction and it is proven in [53] that the multilinear rank of $\widetilde{\mathcal{B}}$ is the same as the multilinear rank of $\mathcal{B}$. These transformed tensors can consequently be used throughout the remaining analysis by directly replacing both tensors and their dimensions.

The computation of the compressed tensors $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ in (A.10) and (A.11) corresponds to the computation of a normalized Tucker-2 decomposition as in (A.4). The compressed tensors equal the core tensors of size (N, N, I_3) of (A.4). The fact that the same matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ can be used for the compression of both $\mathcal{A}$ and $\mathcal{B}$ is an obvious requirement for their decompositions to be similar. Mathematically, the existence of common matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ is implied by (A.8).

It follows from (A.10) and (A.11) that the original tensors $\mathcal{A}$ and $\mathcal{B}$ can be retrieved as follows:

$$\mathcal{A} = \widetilde{\mathcal{A}} \cdot_1 \mathbf{U}_1 \cdot_2 \mathbf{U}_2,$$

$$\mathcal{B} = \widetilde{\mathcal{B}} \cdot_1 \mathbf{U}_1 \cdot_2 \mathbf{U}_2.$$

- **Coinciding spaces of tensor unfoldings in three modes:** In this case, $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ must be compressible to a cubic tensor having full multilinear rank. To verify this, the multilinear rank of $\mathcal{A}$ must be computed. If the mode-n rank of $\mathcal{A}$ is not the same for all three modes, then the discussed theorem cannot be applied but the broader version of [53] explained further in this section is applicable. If the multilinear rank of $\mathcal{A}$ equals (N, N, N) with $N \leq I_1, I_2, I_3$, then (A.9) implies that both tensors can be compressed using the same matrices such that the compressed tensor $\widetilde{\mathcal{A}}$ is cubic and has full multilinear rank.

  To do this compression, the same approach as in the case with two coinciding modes can be used, but in all three modes. This means that orthonormal bases for the row spaces of $\mathbf{A}_{[2,3;1]}, \mathbf{A}_{[1,3;2]}$ and $\mathbf{A}_{[1,2;3]}$ are computed and stored as columns in the matrices $\mathbf{U}_1 \in \mathbb{C}^{I_1 \times N}, \mathbf{U}_2 \in \mathbb{C}^{I_2 \times N}, \mathbf{U}_3 \in \mathbb{C}^{I_3 \times N}$, respectively. The compressed tensors can then be computed as

$$\widetilde{\mathcal{A}} = \mathcal{A} \cdot_1 \mathbf{U}_1^{\mathrm{H}} \cdot_2 \mathbf{U}_2^{\mathrm{H}} \cdot_3 \mathbf{U}_3^{\mathrm{H}}, \tag{A.12}$$

$$\widetilde{\mathcal{B}} = \mathcal{B} \cdot_1 \mathbf{U}_1^{\mathrm{H}} \cdot_2 \mathbf{U}_2^{\mathrm{H}} \cdot_3 \mathbf{U}_3^{\mathrm{H}}. \tag{A.13}$$

  This is equivalent to computing the MLSVD of $\mathcal{A}$ with core tensor $\widetilde{\mathcal{A}} \in \mathbb{C}^{N \times N \times N}$, which has full multilinear rank since the multilinear rank of $\mathcal{A}$ is (N, N, N). Again, the interrelations of both tensors

are not altered since both tensors are compressed using the same factor matrices and (A.9) is assumed to hold [53]. Just as in the previous case, the tensors $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ can then be used throughout the remaining analysis.

2. **Verify coinciding spaces spanned by tensor unfoldings:** One way to characterize the similarity between two subspaces uses principal angles. For R-dimensional subspaces, the principal angles form a set of R minimized angles between both subspaces $\theta_1, \theta_2, \ldots, \theta_R$, ordered from smallest to largest with each angle $\theta_r \in [0, \pi/2]$. The principal angles between two matrices $\mathbf{A}$ and $\mathbf{B}$ are defined recursively, starting with the smallest. For every angle, two vectors $\mathbf{a} \in \text{span}\{\mathbf{A}\}$ and $\mathbf{b} \in \text{span}\{\mathbf{B}\}$ are sought such that the angle between these vectors is minimal under the additional constraint that these vectors must be perpendicular to the vectors that were used for the previous (smaller) principal angles. Mathematically, this can be written as

$$\theta_i \stackrel{\text{def}}{=} \min \left\{ \arccos\left(\frac{|\mathbf{a}^{\mathrm{T}}\mathbf{b}|}{||\mathbf{a}||\,||\mathbf{b}||} \right) \;\middle|\; \mathbf{a} \in \text{span}\{\mathbf{A}\}, \ldots \right.$$

$$\left. \mathbf{b} \in \text{span}\{\mathbf{B}\}, \mathbf{a} \perp \mathbf{a}_j, \mathbf{b} \perp \mathbf{b}_j \quad \forall j \in \{1, \ldots, i-1\} \right\},$$

in which $\mathbf{a}_j$ and $\mathbf{b}_j$ are the vectors corresponding to the angle θ_j. Further mathematical details can be found in [11] and algorithms are given in [80].

Since the principal angles form a set of minimized angles, they will all be zero if the subspaces are exactly the same or will be small if the subspaces are approximately equal, as is often the case in noisy conditions. By contrast, if the subspaces are not equal at all, the principal angles will all be close to $\pi/2$. If only $L < R$ principal angles between two subspaces X and Y are small, this indicates that $X \cap Y$ is an L-dimensional subspace.

To verify the condition that the subspaces coincide, the principal angles must be sufficiently small. The application at hand and the noise level typically determine what “sufficiently small" is. Determining whether subspaces are coinciding based on the principal angles is essentially a machine learning problem, and can be tackled with machine learning algorithms (of which thresholding is a simple example).

3. **Find the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ (and $\mathbf{M}_3$ if applicable):** these matrices can be directly computed from the sets of linear equations

$$\mathbf{A}_{[2,3;1]}\mathbf{M}_1 = \mathbf{B}_{[2,3;1]}, \quad \mathbf{A}_{[1,3;2]}\mathbf{M}_2 = \mathbf{B}_{[1,3;2]},$$

$$\text{and} \quad \mathbf{A}_{[1,2;3]}\mathbf{M}_3 = \mathbf{B}_{[1,2;3]},$$

4. **Compute the eigenvalues and their multiplicities of $\mathbf{M}_1$ and $\mathbf{M}_2$ (and $\mathbf{M}_3$ if applicable):** for this step, standard algorithms can be used.
5. **Determine the structure and relation of $\mathcal{A}$ and $\mathcal{B}$:** by mapping the eigenvalues and their multiplicities from the previous step to the cases presented above, the structure and relation between $\mathcal{A}$ and $\mathcal{B}$ can be determined.

In [53], a more general study of tensor similarity is conducted that also covers the cases of nonsquare tensors and tensors of order higher than 3. This is especially relevant in cases where the results presented in [33] cannot be directly applied, such as for tensors that have different mode-n ranks in each dimension or for higher-order tensors.

For third-order tensors, the main results of [53] can be summarized as follows: consider two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ with $\mathcal{A}$ having full mode-1 and mode-2 rank after compression of both tensors. More explicitly, if $\mathcal{A}$ has mode-1 rank $J_1 < I_1$ and mode-2 rank $J_2 < I_2$, both tensors $\mathcal{A}$ and $\mathcal{B}$ can be compressed such that the transformed tensor $\widetilde{\mathcal{A}} \in \mathbb{C}^{J_1 \times J_2 \times I_3}$ does satisfy the rank conditions. This compression can be done in the same way as explained in equations (A.10) and (A.11), except that J_1 and J_2 may be different in this case. The transformed tensors $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ can then be used throughout the rest of the theorem since their interrelations are not altered by the transformation under condition (A.14) below [53].

Assume the spaces spanned by the matrix unfoldings of $\mathcal{A}$ and $\mathcal{B}$ in two modes are related as follows:

$$\operatorname{span}\left\{\mathbf{B}_{[2,3;1]}\right\} \subseteq \operatorname{span}\left\{\mathbf{A}_{[2,3;1]}\right\} \text{ and } \operatorname{span}\left\{\mathbf{B}_{[1,3;2]}\right\} \subseteq \operatorname{span}\left\{\mathbf{A}_{[1,3;2]}\right\}. \tag{A.14}$$

From (A.14) it follows that there must exist two matrices $\mathbf{M}_1 \in \mathbb{C}^{I_1 \times I_1}$ and $\mathbf{M}_2 \in \mathbb{C}^{I_2 \times I_2}$ such that

$$\mathbf{B}_{[2,3;1]} = \mathbf{A}_{[2,3;1]}\mathbf{M}_1 \text{ and } \mathbf{B}_{[1,3;2]} = \mathbf{A}_{[1,3;2]}\mathbf{M}_2. \tag{A.15}$$

Under these conditions, it is shown in [53] that the tensors $\mathcal{A}$ and $\mathcal{B}$ both admit decompositions into multilinear rank-$(L_{1r}, L_{2r}, \cdot)$ terms which have the same factor matrices but different core tensors:

$$\mathcal{A} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{X}_r^{(1)} \bullet_2 \mathbf{X}_r^{(2)}, \tag{A.16}$$

$$\mathcal{B} = \sum_{r=1}^{R} \lambda_r \mathcal{D}_r \bullet_1 \mathbf{X}_r^{(1)} \bullet_2 \mathbf{X}_r^{(2)}. \tag{A.17}$$

with $\mathcal{D}_r \in \mathbb{C}^{L_{1r} \times L_{2r} \times I_3}$, $\mathbf{X}_r^{(1)} \in \mathbb{C}^{I_1 \times L_{1r}}$ and $\mathbf{X}_r^{(2)} \in \mathbb{C}^{I_2 \times L_{2r}}$. Just as in the previous cases, the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ can be computed directly from the sets of linear equations in (A.15). If these equations have no solution, it means that $\mathcal{B}$ cannot be generated by terms of the decomposition of $\mathcal{A}$ [53]. The value for R in (A.16) and (A.17) can be found as the number of distinct eigenvalues λ_r of $\mathbf{M}_1$ (or $\mathbf{M}_2$). Note that $\mathbf{M}_1$ and $\mathbf{M}_2$ will have the same eigenvalues, but not necessarily the same multiplicities. The (shared) eigenvalues represent the scaling of the different terms in (A.17). The multiplicities of the eigenvalues λ_r of $\mathbf{M}_x$ corresponds to L_{xr}, which is the xth entry in the multilinear rank of the rth term and consequently determines the dimensions of $\mathcal{D}_r$ and $\mathbf{X}_r^{(x)}$.

To use these theorems in practice, the approach outlined before can be used taking into account that the rank conditions are different from the previous cases since nonsquare tensors are allowed after compression. If all conditions are satisfied, both tensors admit the same decomposition with the same factor matrices as outlined in (A.16) and (A.17). The eigenvalues of $\mathbf{M}_1$ (or $\mathbf{M}_2$) can then be used to determine the number of terms and the scaling of the terms in both decompositions. Note that the procedure can also be used for approximate decompositions, for instance when noise is present. In this case, equation (A.15) can be solved in least-squares sense and the average of eigenvalues λ_{1r} and λ_{2r} can be used to estimate λ_r. The values L_{1r} and L_{2r} can then be estimated by assessing how close the different eigenvalues λ_{nr} are to each of the estimated eigenvalues λ_r [53].

For further generalizations, mathematical specifics and proofs we refer to [53].

A.4.3 Similarity for specific decompositions

CPD similarity in two modes

In the previous section, no prior knowledge was available on the type of decomposition $\mathcal{A}$ and $\mathcal{B}$ admit. Here, it is assumed that both tensors admit a rank-R CPD, which simplifies the approach and computations. Moreover, only two modes of third-order tensors are compared. This is motivated by the observation that in several applications, such as emitter localization in telecommunications, the information of interest is contained in just two modes of a tensor decomposition [145]. In emitter localization for example, all information on the emitter location is contained in the first two factor matrices after computing a tensor decomposition, while the factor matrix in the third mode contains information about the source signals [14].

As stated in the problem statement in Section A.4.1, the aim is to determine

whether $\mathcal{A}$ and $\mathcal{B}$ are essentially equal in two modes. Mathematically, this translates to following problem statement: consider two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ that both admit a rank-R CPD:

$$\begin{aligned} \mathcal{A} &= \left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \right]\!\right], \\ \mathcal{B} &= \left[\!\left[\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)} \right]\!\right]. \end{aligned} \tag{A.18}$$

The goal is to compare the matrices $\mathbf{U}^{(1)}$ and $\mathbf{V}^{(1)} \in \mathbb{C}^{I_1 \times R}$, and $\mathbf{U}^{(2)}$ and $\mathbf{V}^{(2)} \in \mathbb{C}^{I_2 \times R}$ up to the trivial indeterminacies of the CPD. This comparison can be done for the full factor matrices ("Full similarity in two modes") or for a subset of the CPD terms ("Partial similarity in two modes").

In this formulation, the first two out of three factor matrices are compared. If one wishes to compare other factors, the modes of the tensors need to be permuted first such that the modes of interest are the first two modes.

Full similarity in two modes The heart of the approach for full similarity in two modes lies in the following theorem [145]:

Theorem 9. *Let two tensors $\mathcal{A}, \mathcal{B}$ admit a rank-R CPD as shown in* (A.18)*. If following conditions hold:*

(i) $\quad$ $\mathbf{U}^{(3)}$ *and* $\mathbf{V}^{(3)}$ *have full column rank,*

(ii) $\quad$ $\mathcal{C}_2\left(\mathbf{U}^{(1)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(2)}\right)$ *has full column rank,*

then $\mathcal{A}$ and $\mathcal{B}$ are essentially equal in the first two modes if and only if $span\left\{\mathbf{A}_{[1,2;3]}\right\} = span\left\{\mathbf{B}_{[1,2;3]}\right\}$.

In this theorem, $\mathcal{C}_2(\mathbf{K})$ is defined as the second compound matrix of an $I \times R$ matrix $\mathbf{K}$, which is the $\binom{I}{2} \times \binom{R}{2}$ matrix containing the determinants of all 2×2 submatrices of $\mathbf{K}$ [49]. The determinants are arranged such that the submatrix index sets are in lexicographic order. The intuition for the theorem is as follows: the conditions (i) and (ii) actually correspond to the uniqueness conditions for the CPD of $\mathcal{A}$. This implies that there exist no other rank-1 terms in the space $\text{span}\left\{\mathbf{A}_{[1,2;3]}\right\}$ apart from the rank-1 terms of the first two modes of $\mathcal{A}$. Consequently, if $\text{span}\left\{\mathbf{A}_{[1,2;3]}\right\} = \text{span}\left\{\mathbf{B}_{[1,2;3]}\right\}$, the rank-1 terms in the first two modes of $\mathcal{B}$ must be the same up to scaling and permutation. The mathematical details of the proof can be found in [145]. The practical implications of this theorem are as follows: if the conditions hold, one can simply check whether the columns of both unfolded tensors span the same

space. If this is the case, the factor $\mathbf{U}^{(1)}$ equals $\mathbf{V}^{(1)}$ up to column scaling and permutation, and $\mathbf{U}^{(2)}$ equals $\mathbf{V}^{(2)}$ up to column scaling and permutation.

Though the approach is conceptually simple, the details can be challenging. Below, we outline the steps one can follow to compare the factors in two modes of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ that both admit a rank-R CPD. These steps are largely based on the approach outlined in [145].

1. **Verify the conditions:** First, the factor matrices in the third mode of both tensors ($\mathbf{U}^{(3)}$ and $\mathbf{V}^{(3)}$) need to have full column rank. Second, the matrix $\mathcal{C}_2\left(\mathbf{U}^{(1)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(2)}\right)$ needs to have full column rank as well. The challenge in verifying these conditions is that the factor matrices are not explicitly available. Indeed, the whole purpose of the algorithm is to compare tensor factors without having to explicitly compute them. Following approaches can be used to verify the conditions:

   - The full column rank condition of a factor matrix (e.g., $\mathbf{U}^{(3)} \in \mathbb{C}^{I_3 \times R}$) can be verified as follows: consider the unfolded tensor $\mathbf{A}_{[1,2;3]}$ and check whether its rank equals R. Because this unfolding can be written as $\mathbf{A}_{[1,2;3]} = \left(\mathbf{U}^{(1)} \odot \mathbf{U}^{(2)}\right) {\mathbf{U}^{(3)}}^{\mathrm{T}} \in \mathbb{C}^{I_1 I_2 \times I_3}$, this will be the case if $\mathbf{U}^{(3)}$ has full column rank and $R \leq I_3$ and $R \leq I_1 I_2$.
   - To verify whether $\mathcal{C}_2\left(\mathbf{U}^{(1)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(2)}\right)$ has full column rank, one can turn to [30] where it is shown that this condition is the same as requiring that the tensors $\left\{\Phi\left(\mathbf{u}_t^{(1)}{\mathbf{u}_t^{(2)}}^{\mathrm{T}}, \mathbf{u}_v^{(1)}{\mathbf{u}_v^{(2)}}^{\mathrm{T}}\right)\right\}_{1 \leq t < v \leq R}$ are linearly independent. In this condition, $\Phi(\cdot,\cdot)$ is a bilinear mapping that maps two matrices $\mathbf{M}, \mathbf{N} \in \mathbb{C}^{I \times J}$ to a fourth-order tensor $\Phi(\mathbf{M}, \mathbf{N}) \in \mathbb{C}^{I \times I \times J \times J}$ given by

     $$(\Phi(\mathbf{M}, \mathbf{N}))_{ijkl} = m_{ik} n_{jl} + m_{jl} n_{ik} - m_{il} n_{jk} - m_{jk} n_{il}. \quad \text{(A.20)}$$

     To verify that the tensors $\left\{\Phi\left(\mathbf{u}_t^{(1)}{\mathbf{u}_t^{(2)}}^{\mathrm{T}}, \mathbf{u}_v^{(1)}{\mathbf{u}_v^{(2)}}^{\mathrm{T}}\right)\right\}_{1 \leq t < v \leq R}$ are linearly independent without knowledge of the factor matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$, first construct a basis $\mathbf{E}$ for $\mathbf{A}_{[1,2;3]} = \left(\mathbf{U}^{(1)} \odot \mathbf{U}^{(2)}\right) {\mathbf{U}^{(3)}}^{\mathrm{T}} \in \mathbb{C}^{I_1 I_2 \times I_3}$ by selecting the left singular vectors corresponding to the R dominant singular values and storing these as columns in $\mathbf{E} \in \mathbb{C}^{I_1 I_2 \times R}$. Next, construct the matrices $\mathbf{E}_r \in \mathbb{C}^{I_1 \times I_2}$ by matricizing each column such that

     $$(\mathbf{E}_r)_{i_1 i_2} = e_{(i_1 - 1) I_2 + i_2, r} \qquad \forall i_1, i_2, r.$$

It now follows from the derivation in [30] that if the tensors $\{\Phi(\mathbf{E}_r, \mathbf{E}_s)\}_{1\leq r\leq s\leq R}$ are linearly independent, then the tensors $\left\{\Phi\left(\mathbf{u}_t^{(1)}\mathbf{u}_t^{(2)^{\mathrm{T}}}, \mathbf{u}_v^{(1)}\mathbf{u}_v^{(2)^{\mathrm{T}}}\right)\right\}_{1\leq t<v\leq R}$ are linearly independent as well and it consequently holds that $\mathcal{C}_2\left(\mathbf{U}^{(1)}\right)\odot\mathcal{C}_2\left(\mathbf{U}^{(2)}\right)$ has full column rank. To verify that the tensors $\{\Phi(\mathbf{E}_r, \mathbf{E}_s)\}_{1\leq r\leq s\leq R}$ are linearly independent, first compute them via (A.20). Next, vectorize the results to obtain vectors $\mathbf{p}_{rs}\in\mathbb{C}^{I_1^2I_2^2}$ and stack these vectors in a matrix $\mathbf{P}\in\mathbb{C}^{I_1^2I_2^2\times R(R+1)/2}$. If $\mathbf{P}$ has full column rank, the tensors $\{\Phi(\mathbf{E}_r, \mathbf{E}_s)\}_{1\leq r\leq s\leq R}$ are linearly independent and it follows that $\mathcal{C}_2\left(\mathbf{U}^{(1)}\right)\odot\mathcal{C}_2\left(\mathbf{U}^{(2)}\right)$ has full column rank..

A different approach to validate the conditions is based on generic properties, which are easier to check but only hold for "almost all" cases.

- For the first condition, which states that $\mathbf{U}^{(3)}\in\mathbb{C}^{I_3\times R}$ and $\mathbf{V}^{(3)}\in\mathbb{C}^{I_3\times R}$ must have full column rank, the property can be used that states that a generic matrix has full column rank if it has at least as many rows as columns. Consequently, the first condition is satisfied if $I_3\geq R$.
- For the second condition, it is stated in [50] that a matrix $\mathcal{C}_2(\mathbf{X})\odot\mathcal{C}_2(\mathbf{Y})$ generically has full column when it has at least as many rows as columns. In our case, the matrix $\mathcal{C}_2\left(\mathbf{U}^{(1)}\right)\odot\mathcal{C}_2\left(\mathbf{U}^{(2)}\right)$ has dimensions $\binom{I_1}{2}\binom{I_2}{2}\times\binom{R}{2}$, which means that the condition is generically satisfied if $\binom{I_1}{2}\binom{I_2}{2}\geq\binom{R}{2}$.

2. **Determine the number of components:** In many applications, the rank R of the tensors is not known in advance. Determining the rank of a tensor is an NP-hard problem and often requires decomposing the tensor trying various ranks. Since the approach presented here aims to avoid computing tensor decompositions, prior knowledge or heuristics are needed. In the simplest case, the rank is known from the application. If no prior knowledge is available, one can resort to estimates. As mentioned earlier, a lower bound on the tensor rank is the number of significant singular values of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$, capturing most of the information of both matrices. Under the assumptions of Theorem 9, these matrices will have rank R in the noiseless case. If noise is present, there will be R dominant singular values and the other singular values are expected to be significantly smaller.

3. **Find a basis for the dominant column spaces of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$:** The subspace spanned by the columns of $\mathbf{A}_{[1,2;3]}$ and the number of

components R are known. In this step, the goal is to construct a basis for the R-dimensional dominant subspace of $\mathbf{A}_{[1,2;3]}$. One way to do this is by computing the left singular vectors of $\mathbf{A}_{[1,2;3]}$ corresponding to the R largest principal values. Store these left singular vectors as the columns in a matrix $\mathbf{X} \in \mathbb{C}^{I_1 I_2 \times R}$. The same approach can be used to extract a basis $\mathbf{Y} \in \mathbb{C}^{I_1 I_2 \times R}$ for the dominant subspace of $\mathbf{B}_{[1,2;3]}$.

4. **Compute the principal angles between both subspaces:** In the previous step, the bases $\mathbf{X} \in \mathbb{C}^{I_1 I_2 \times R}$ and $\mathbf{Y} \in \mathbb{C}^{I_1 I_2 \times R}$ were constructed for both the subspaces spanned by the columns of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$, respectively. The principal angles between these two bases can then be computed as explained in Section A.4.2.

5. **Classify the subspaces as equal or not equal:** Based on the R principal angles, classify the subspaces must be classified as “equal" or “not equal" based on whether the principal angles are sufficiently small.

Partial similarity in two modes The theorem and procedure in the previous section aims at determining whether all terms of two CPDs are essentially equal in two modes. Here, partial similarity is considered, where only a subset of the terms of both CPDs are essentially equal in two modes. Again, the approach is based on a central theorem:

Theorem 10. *Consider two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ admitting a rank-R CPD as defined in* (A.18)*. If*

(i) $\mathbf{U}^{(3)}$ *and* $\mathbf{V}^{(3)}$ *have full column rank R,*

(ii) $\mathcal{C}_2\left(\mathbf{U}^{(2)}\right) \odot \mathcal{C}_2\left(\mathbf{U}^{(1)}\right)$ *has full column rank,*

(iii) $\mathcal{C}_2\left(\mathbf{V}^{(2)}\right) \odot \mathcal{C}_2\left(\mathbf{V}^{(1)}\right)$ *has full column rank,*

(iv) $\mathit{span}\left\{\mathbf{A}_{[1,2;3]}\right\} \cap \mathit{span}\left\{\mathbf{B}_{[1,2;3]}\right\}$ *is Q-dimensional and admits a unique (unfolded) rank-Q CPD,*

then there are Q essentially equal terms in two modes if and only if Q principal angles between the unfolded tensors $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$ are zero, with $Q \leq R$.

The approach for partial similarity is identical to the approach outlined under “full similarity in two modes”, apart from two differences:

- In the first step, the conditions of Theorem 10 must be verified. To do this, the same techniques as before can be used for the first three conditions.

  The fourth condition states that the Q-dimensional space $\left(\text{span}\left\{\mathbf{A}_{[1,2;3]}\right\} \cap \text{span}\left\{\mathbf{B}_{[1,2;3]}\right\}\right)$ must admit a rank-Q CPD. Say a basis of $\left(\text{span}\left\{\mathbf{A}_{[1,2;3]}\right\} \cap \text{span}\left\{\mathbf{B}_{[1,2;3]}\right\}\right)$ is stacked in $\mathbf{W} \in \mathbb{C}^{I_1 I_2 \times Q}$. This matrix is to be seen as the unfolding $\mathbf{W}_{[1,2;3]}$ of a third-order tensor $\mathcal{W} \in \mathbb{C}^{I_1 \times I_2 \times Q}$. Now a rank-$Q$ CPD of $\mathcal{W}$ needs to be computed to check whether it exists/admits a good approximation of $\mathcal{W}$ and whether it is unique (e.g., the uniqueness conditions in [52, 50, 110, 84]). This means that it is not possible to compare partial similarity in two modes without computing at least this decomposition, which risks issues with ill-conditioning.

- In the last step in which the subspaces are classified based on the principal angles, only the Q smallest principal angles must be considered for the classification problem.

If two tensors are partially equal in two modes, the common terms can be extracted without computing the full tensor decompositions of both tensors. This is especially advantageous when both tensors admits high-rank CPDs with only a few terms in common. The interested reader is refered to [145] for a step-by-step approach on how to do this.

BTD in rank-$(L, L, 1)$ terms similarity in two modes

This section assumes that both tensors $\mathcal{A}$ and $\mathcal{B}$ admit a BTD in R multilinear rank-$(L, L, 1)$ terms, which, compared to Section A.4.2 simplifies the approach and computations but, compared to the CPD case in Section A.4.3, allows tensor similarity in situations where the rank-1 assumption is too strict. The aim here is to determine whether both tensors are essentially equal in two modes. Mathematically, this translates to following problem statement. Consider two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ that both admit a BTD in R multilinear rank-$(L, L, 1)$ terms:

$$\mathcal{A} = \sum_{r=1}^{R} \left(\mathbf{U}_r^{(1)} \mathbf{U}_r^{(2)^{\mathrm{T}}}\right) \circ \mathbf{u}_r^{(3)}, \tag{A.22}$$

$$\mathcal{B} = \sum_{r=1}^{R} \left(\mathbf{V}_r^{(1)} \mathbf{V}_r^{(2)^{\mathrm{T}}}\right) \circ \mathbf{v}_r^{(3)}, \tag{A.23}$$

with $\mathbf{U}_r^{(1)}, \mathbf{V}_r^{(1)} \in \mathbb{C}^{I_1 \times L}$ and $\mathbf{U}_r^{(2)}, \mathbf{V}_r^{(2)} \in \mathbb{C}^{I_2 \times L}$. The goal is to assess whether the low-rank matrices $\mathbf{U}_r^{(1)}{\mathbf{U}_r^{(2)}}^{\mathrm{T}}$ and $\mathbf{V}_r^{(1)}{\mathbf{V}_r^{(2)}}^{\mathrm{T}}$ are the same up to permutation and scaling.

The heart of the approach lies in following theorem, adapted from [145]:

Theorem 11. *Consider two tensors $\mathcal{A}, \mathcal{B}$ admitting a BTD in R rank-$(L, L, 1)$ terms as defined in* (A.22) *and* (A.23)*. If following conditions hold*

(i) $\mathbf{U}^{(3)}$ *and* $\mathbf{V}^{(3)}$ *have full column rank,*

(ii) *The BTD of $\mathcal{A}$ is essentially unique,*

then $\mathcal{A}$ and $\mathcal{B}$ are essentially equal in the first two modes if and only if $\mathit{span}\left\{\mathbf{A}_{[1,2;3]}\right\} = \mathit{span}\left\{\mathbf{B}_{[1,2;3]}\right\}$.

In practice, following steps can be followed [145]:

1. **Verifying the conditions:** Since the factor matrices are not readily available, alternative approaches are needed to verify the conditions. The factor matrices in the third mode of both tensors ($\mathbf{U}^{(3)}$ and $\mathbf{V}^{(3)}$) need to have full column rank, which can be checked using the same approach presented in Section A.4.3. To determine whether the BTD of $\mathcal{A}$ is essentially unique, deterministic uniqueness conditions can be found in [54], though these require access to (some algebraic properties of) the factors of $\mathcal{A}$. In some cases, these properties can be inferred directly from $\mathcal{A}$, without computing the decomposition. For instance, if the multilinear rank of $\mathcal{A}$ equals (LR, LR, R), then the BTD in (A.22) is necessarily essentially unique. Generic uniqueness conditions which only use the dimensions of $\mathcal{A}$ and its factors can be found in [31, 54].
2. **Determine the number of components:** As before, (a lower bound on) the number of components is given by the number of dominant singular values of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$. If both conditions of Theorem 11 hold, these matrices will have rank R in the noiseless case. If noise is present, there will be R dominant singular values and the other singular values are expected to be significantly smaller.
3. **Find a basis for the dominant column spaces of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$:** In this step, the same techniques as in Section A.4.3 can be used.
4. **Compute the principal angles between both subspaces:** As in Section A.4.2, principal angles can be used as a similarity measure for the subspaces span $\left\{\mathbf{A}_{[1,2;3]}\right\}$ and span $\left\{\mathbf{A}_{[1,2;3]}\right\}$.

5. **Classify the subspaces as equal or not equal:** This step is identical to the approach outlined in Section A.4.3.

A.5 Similarity in chemometrics

This section aims to clarify the use of the procedures and theorems given in the previous sections by applying them to a simple chemometrics example. Consider the amino acids fluorescence dataset (`http://www.models.life.ku.dk/amino_acid_fluo`), which consists of five simple laboratory-made samples. Each of these samples consists of different amounts of tyrosine, tryptophan and phenylalanine dissolved in phosphate-buffered water. Two samples are mixtures of all components, while the other three samples are measurements of the separate analytes. The samples were measured by fluorescence with excitation wavelengths of $240 - 300$ nm and emmision wavelengths of $250 - 450$ nm with 1 nm intervals. For more technical details, we refer the reader to [15] or the URL given earlier.

A.5.1 Comparison of sets of mixtures without prior information

In Section A.4.2, a procedure was outlined to assess the relationship between two tensors without any prior knowledge on their decompositions. To illustrate this in a chemometrics context, two sets of four mixtures are artificially created using the excitation-emission spectra of the pure analytes in the amino acids fluorescence dataset (tyrosine, tryptophan and phenylalanine). The resulting excitation-emission matrices of each mixture are subsequently stacked per mixture set as frontal slices in two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{201\times 61\times 4}$. The concentrations of the analytes in each frontal slice are given in Table A.1.

To compare $\mathcal{A}$ and $\mathcal{B}$ without making any assumption on their decompositions, the procedure outlined in Section A.4.2 can be followed:

1. **Verify the rank conditions:** The multilinear rank of both tensors can be determined by computing the dominant singular values of their matrix unfoldings. For $\mathcal{A}$, the largest three singular values in each mode capture most of the information, as illustrated in Figure A.8. This implies that the multilinear rank of $\mathcal{A}$ is $(3, 3, 3)$. Analogously, it can be shown that $\mathcal{B}$ has multilinear rank $(3, 3, 3)$ as well. Before any compression of the tensors can be done, it must be verified whether their matrix unfoldings span coinciding spaces.

Table A.1: Concentrations of the analytes Tyrosine, Tryptophan and Phenylalanine in each mixture, represented by the frontal slices of $\mathcal{A}$ and $\mathcal{B}$ which contain the excitation-emission spectra of the mixtures.

	Tyr	Tryp	Phe
$\mathbf{A}_{::1}$	$0.0267 \cdot 10^{-3}$	$0.0133 \cdot 10^{-3}$	$0.1800 \cdot 10^{-3}$
$\mathbf{A}_{::2}$	$0.0267 \cdot 10^{-3}$	$0.0266 \cdot 10^{-3}$	$0.2700 \cdot 10^{-3}$
$\mathbf{A}_{::3}$	$0.0320 \cdot 10^{-3}$	$0.0067 \cdot 10^{-3}$	$0.3600 \cdot 10^{-3}$
$\mathbf{A}_{::4}$	$0.0107 \cdot 10^{-3}$	$0.0100 \cdot 10^{-3}$	$0.7200 \cdot 10^{-3}$
$\mathbf{B}_{::1}$	$0.0133 \cdot 10^{-3}$	$0.0266 \cdot 10^{-3}$	$0.1350 \cdot 10^{-3}$
$\mathbf{B}_{::2}$	$0.0133 \cdot 10^{-3}$	$0.0532 \cdot 10^{-3}$	$0.2025 \cdot 10^{-3}$
$\mathbf{B}_{::3}$	$0.0160 \cdot 10^{-3}$	$0.0133 \cdot 10^{-3}$	$0.2700 \cdot 10^{-3}$
$\mathbf{B}_{::4}$	$0.0053 \cdot 10^{-3}$	$0.0200 \cdot 10^{-3}$	$0.5400 \cdot 10^{-3}$

2. **Verify coinciding spaces spanned by tensor unfoldings:** To compute the principal angles, standard algorithms can be used (e.g., [80]). Table A.2 shows the five largest principal angles for each of the equations in (A.9). The principal angles are extremely low for the comparison in two modes. For the comparison of span $\left\{\mathbf{A}_{[1,3;2]}\right\}$ and span $\left\{\mathbf{B}_{[1,3;2]}\right\}$, the principal angles are significantly larger. However, noise must be taken into account since this example is based on actual excitation-emission spectroscopy measurements of analytes. With this in mind, the largest principal angle (0.6478) is still small compared to the value for perpendicular spaces ($\pi/2$). Consequently, it is reasonable to conclude that the spaces spanned by the matrix unfoldings in three modes can be considered equal, i.e., (A.9) holds.

   Since it has been shown that $\mathcal{A}$ is compressible to a $(3,3,3)$-tensor with full multilinear rank and (A.9) holds, tensors $\mathcal{A}$ and $\mathcal{B}$ can be compressed to $\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}} \in \mathbb{R}^{3\times3\times3}$ as in (A.12) and (A.13). It can be verified that the principal angles between the spaces spanned by the matrix unfoldings of $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ are still sufficiently small to be considered equal.

3. **Find the matrices $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$:** This step is trivial.

4. **Compute the eigenvalues and their multiplicities of $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$:** The eigenvalues of all three matrices are given in Table A.3. The table shows that the matrices have clearly distinct eigenvalues. Note that the eigenvalues of the different matrices lie close to eachother but are not exactly the same since noise is present in the measurement data and the compressed tensors are not perfect mixtures.

Table A.2: Largest principal angles for the comparison of the spaces spanned by the matrix unfoldings of $\mathcal{A}$ and $\mathcal{B}$ in each mode.

	$\text{span}\left\{\mathbf{A}_{[2,3;1]}\right\}$ $\overset{?}{=}$ $\text{span}\left\{\mathbf{B}_{[2,3;1]}\right\}$	$\text{span}\left\{\mathbf{A}_{[1,3;2]}\right\}$ $\overset{?}{=}$ $\text{span}\left\{\mathbf{B}_{[1,3;2]}\right\}$	$\text{span}\left\{\mathbf{A}_{[1,2;3]}\right\}$ $\overset{?}{=}$ $\text{span}\left\{\mathbf{B}_{[1,2;3]}\right\}$
Principal angles	$1.168 \cdot 10^{-9}$	0.4759	$0.838 \cdot 10^{-13}$
	$1.251 \cdot 10^{-9}$	0.4885	$0.948 \cdot 10^{-13}$
	$1.629 \cdot 10^{-9}$	0.5346	$7.920 \cdot 10^{-13}$
	$1.917 \cdot 10^{-9}$	0.5753	
	$2.567 \cdot 10^{-9}$	0.6478	

Table A.3: Largest principal angles for the comparison of the spaces spanned by the matrix unfoldings of $\mathcal{A}$ and $\mathcal{B}$ in each mode.

$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$
0.5518	1.8455	0.7418
0.7647	0.5557	0.5524
1.6818	0.7026	1.7090

5. **Determine the structure and relation of $\mathcal{A}$ and $\mathcal{B}$:** Since all eigenvalues of $\mathbf{M}_1$ (or $\mathbf{M}_2$ or $\mathbf{M}_3$) are distinct, it follows that $\mathcal{A}$ and $\mathcal{B}$ admit a rank-3 CPD consisting of the same (possibly differently scaled) terms. Each of the terms represent one of the pure analytes in this example. More specifically, the factor vector in the first mode of each term contains the excitation information, the second the emission information and the third the concentrations. Since both tenors consist of the same terms, the concentrations of each analyte in the mixtures of $\mathcal{B}$ should be a scaled version of the concentrations in the mixtures of $\mathcal{A}$. Indeed, it can be verified from Table A.1 that the mixtures in the frontal slices of $\mathcal{B}$ contain 0.5 times the concentration of Tyrosine in $\mathcal{A}$, 2 times the concentration of Tryptophan and 0.75 times the concentration of Phenylalanine. These scaling factors are retrieved as the eigenvalues of $\mathbf{M}_1$ (or $\mathbf{M}_2$ or $\mathbf{M}_3$), but in this case only approximately because of the influence of noise.

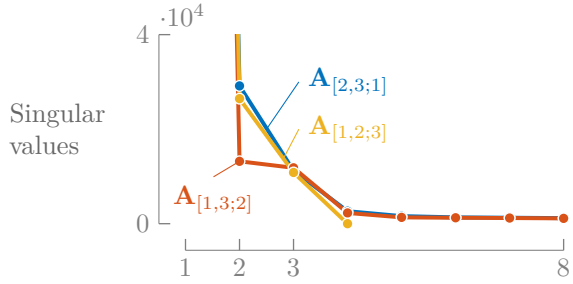


Figure A.8: The first three singular values of the matrix unfoldings $\mathbf{A}_{[2,3;1]}$, $\mathbf{A}_{[1,3;2]}$ and $\mathbf{A}_{[1,2;3]}$ contain most of the information.

A.5.2 Comparison of analytes in mixtures (with prior information)

In this example, the approach of Section A.4.3 is illustrated in two chemometrics applications as was done in [145], though more details and a step-by-step outline are provided in this chapter. Specifically, the goal is to determine whether two amino acid fluorescence mixture samples contain the same analytes.

In a first experiment, the two samples of the amino acids fluorescence dataset containing mixtures are selected. Each of these samples is represented by a matrix of size 61×201. To transform these matrices to tensors, Löwnerization is used along the emission mode as explained in Section A.3.4. This yields two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{101\times 100\times 61}$, which both admit a BTD in rank-$(2,2,1)$ terms, which means that each of the pure analytes contributes one multilinear rank-$(2,2,1)$ term to the tensors [46]. In section A.3.4 it was explained how the number of components and the multilinear rank of the terms can be determined based on the tensors $\mathcal{A}$ and $\mathcal{B}$.

To compare the underlying analytes of both mixtures, the steps outlined in Section A.4.3 are followed:

1. **Verifying the conditions:** Because the tensors admit BTDs in multilinear rank-$(2,2,1)$ terms, the conditions of Theorem 11 need to be verified.

    - The first condition states that the factors in the third mode must have full column rank. To verify this, the dominant singular values of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$ are computed. Figure A.9 shows the eight largest principal angles for both matrices. Most of the information seems to be captured by the first three singular vectors. For this

reason, the number of components is estimated to be equal to $R = 3$ and it immediately follows that the factors in the third mode must have full column rank.

- The second condition says that the BTD of $\mathcal{A}$ must be essentially unique. For this, Theorem 4.1 in [31] can be used, which states that this BTD is generically unique if $\min(I, J) \geq RL$ and the third factor does not have proportional columns. The number of components was computed to be $R = 3$ based on the dominant singular values of $\mathbf{A}_{[1,2;3]}$, from which it follows that the third factor does not have proportional columns. To determine L, one can compute the number of dominant singular values of $\mathbf{A}_{[2,3;1]}$, which equals RL if $\min(I, J) \geq RL$. In this experiment, it can be verified that $\mathbf{A}_{[2,3;1]}$ has 6 dominant singular values. Since the other dimensions are known: $I = 101$, $J = 100$ and $R = 3$, it follows that $L = 2$ and that $\min(I, J) \geq RL$ effectively holds. It thus follows that the BTD of $\mathcal{A}$ is essentially unique.

2. **Determine the number of components:** The number of components was already computed to be $R = 3$ when verifying the conditions. This value means that the considered mixtures are made up out of three pure analytes, each contributing one rank-$(2, 2, 1)$ term to the tensor.

3. **Find a basis for the dominant column spaces of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$:** This can be done by taking the left singular vectors that correspond to the three largest singular values and storing them in the matrices $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(2)} \in \mathbb{C}^{10100 \times 3}$.

4. **Compute the principal angles between both subspaces:** For this step, an implementation of the algorithm described in [80] was used. The computed principal values are 0.037, 0.064 and 0.328.

5. **Classify the subspaces as equal or not equal:** To classify whether the subspaces are equal or not, the principal angles are examined in more detail. If the subspaces are perpendicular, all principal angles are $\pi/2$. In this application, noise is present and the largest principal angle (0.328) is still reasonably low. For our example application, it can be concluded that the subspaces can be considered equal.

Because the conditions in Theorem 11 are valid and the (dominant) subspaces of the unfolded tensors are equal, it follows that $\mathcal{A}$ and $\mathcal{B}$ are essentially equal in the first two modes. This means that the block terms making up both tensors are the same up to scaling and permutation. Since each of these block terms represents a pure analyte, it can be concluded that both samples are mixtures

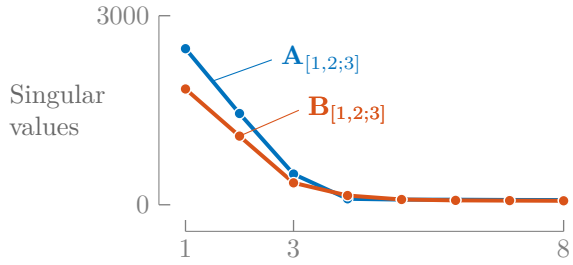


Figure A.9: The first three singular values of $\mathbf{A}_{[1,2;3]}$ and $\mathbf{B}_{[1,2;3]}$ contain most of the information. Adopted with changes from [145].

of the same pure analytes. The description of the data set confirms this because both samples have been obtained by mixing different amounts of the same three amino acids: Tyrosine, Tryptophan and Phenylalanine.

Apart from the mixtures above, the dataset also contains measurements of the three pure analytes. In a final experiment, these measurements were used to artificially create several mixtures consisting of two analytes. The approach of [145] is followed and following mixtures have been constructed:

$$\mathrm{EE}\,(\mathrm{Mix1}) = \mathrm{EE}\,(\mathrm{Trp}) + \mathrm{EE}\,(\mathrm{Tyr})\,, \tag{A.25}$$

$$\mathrm{EE}\,(\mathrm{Mix2}) = 0.4\ \mathrm{EE}\,(\mathrm{Tyr}) + 0.8\ \mathrm{EE}\,(\mathrm{Phe})\,,$$

$$\mathrm{EE}\,(\mathrm{Mix3}) = 0.8\ \mathrm{EE}\,(\mathrm{Trp}) + 0.4\ \mathrm{EE}\,(\mathrm{Tyr})\,, \tag{A.26}$$

in which EE (x) is a shorthand to denote the excitation-emission matrix of x. The construction of the mixtures shows that Mix1 and Mix3 contain the same pure analytes. Let us now use tensor similarity to verify this statement. A step-by-step outline is given below:

First, the data needs to be tensorized since only three separate excitation-emission matrices are available. For this, Löwnerization is used as in the previous experiment to obtain tensors $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, which all have dimensions $101 \times 100 \times 61$. Next, the procedure to compare tensors that admit a BTD in rank-$(2, 2, 1)$ terms can be used, just as before:

1. **Verifying the conditions:** Because the tensors admit BTDs in rank-$(2, 2, 1)$ terms, the conditions of Theorem 11 must be verified. It can be verified in exactly the same way as in the previous experiment that both conditions hold for any two tensors of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ one wishes to compare.

Table A.4: The principal angles clearly indicate which mixtures consist of the same components. Adopted from [145].

Mixtures to compare	Principal angles (rad)	
Mix1 and Mix2	0.0664	1.3032
Mix1 and Mix3	0.0217	0.0674
Mix2 and Mix3	0.1198	1.2892

2. **Determine the number of components:** By computing the number of dominant singular values of $\mathbf{A}_{[1,2;3]}$, $\mathbf{B}_{[1,2;3]}$ and $\mathbf{C}_{[1,2;3]}$, it can be verified that all three tensors lead to $R = 2$ as the number of components. This is in line with the construction of the mixtures out of two pure analytes.
3. **Find a basis for the dominant column spaces of $\mathbf{A}_{[1,2;3]}$, $\mathbf{B}_{[1,2;3]}$ and $\mathbf{C}_{[1,2;3]}$:** This is done by taking the left singular vectors that correspond to the two largest singular values.
4. **Compute the principal angles between the subspaces:** To compute the principal angles an implementation of the algorithm described in [80] is used. Table A.4 shows the principal angles between all mixtures.
5. **Classify the subspaces as equal or not equal:** Table A.4 shows that all principal angles between Mix1 and Mix3 are small. The other combinations have one principal angle that is significantly larger. From these observations, it follows that the subspaces related to Mix1 and Mix3 can be considered equal.

Because the conditions of Theorem 11 are verified and the subspaces of the first and third mixture can be considered equal, it follows that Mix1 and Mix3 consist of the same pure analytes. This is indeed confirmed by the construction of the mixtures in (A.25) - (A.26), which shows that the same pure analytes have been used to create Mix1 and Mix3. By contrast, the principal angles clearly showed that Mix2 does not consist of the same pure analytes. Though it is not part of any of the theorems presented in this chapter, note that one of the principal angles between Mix2 and the other mixtures is small since one of the pure analytes is shared.

A.6 Conclusion

Tensors have been widespread in chemometrics for decades. In this chapter, state-of-the-art results on tensor similarity have been presented. The specific

properties of tensors allow one to extract information on the decompositions of two tensors, without having to compute these decompositions. Results and algorithms have been discussed for tensors admitting CPDs and BTDs in multilinear rank-$(L, L, 1)$ terms, and for tensors of which the decomposition is not known a priori. The procedure and results have been illustrated on a simple chemometrics example as well.

Bibliography

[1] Abed-Meraim, K., and Belouchrani, A. Algorithms for joint block diagonalization. In *Signal Processing Conference, 2004 12th European* (2004), IEEE, pp. 209–212.

[2] Abed-Meraim, K., Qiu, W., and Hua, Y. Blind system identification. *Proc. IEEE 85*, 8 (1997), 1310–1322.

[3] Acar, E., Dunlavy, D. M., Kolda, T. G., and Mørup, M. Scalable tensor factorizations for incomplete data. *Chemometr. Intell. Lab. 106*, 1 (2011), 41–56.

[4] Acar, E., Kolda, T. G., and Dunlavy, D. M. All-at-once optimization for coupled matrix and tensor factorizations. *KDD Workshop on Mining and Learning with Graphs* (2011).

[5] Acar, E., Rasmussen, M. A., Savorani, F., Næs, T., and Bro, R. Understanding data fusion within the framework of coupled matrix and tensor factorizations. *Chemometr. Intell. Lab. 129* (2013), 53–63.

[6] Accenture, and Mighty AI. Autonomous vehicles: the race is on, 2018. [Online; visited Sep. 18, 2018].

[7] Anandkumar, A., Ge, R., Hsu, D. J., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research 15*, 1 (2014), 2773–2832.

[8] Baggerly, K. A., Morris, J. S., and Coombes, K. R. Reproducibility of seldi-tof protein patterns in serum: comparing datasets from different experiments. *Bioinformatics 20*, 5 (2004), 777–785.

[9] Baxter, P., and McWhirter, J. Blind signal separation of convolutive mixtures. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers* (2004), vol. 1, IEEE, pp. 124–128.

[10] Belouchrani, A., Abed-Meraim, K., Cardoso, J.-F., and Moulines, E. A Blind Source Separation Technique Using Second-Order Statistics. *IEEE Trans. Signal Process. 45*, 2 (1997), 434–444.

[11] Björck, A., and Golub, G. H. Numerical methods for computing angles between linear subspaces. *Mathematics of computation 27*, 123 (1973), 579–594.

[12] Bousbia-Salah, H., Belouchrani, A., and Abed-Meraim, K. Blind separation of convolutive mixtures using joint block diagonalization. In *Sixth International Symposium on Signal Processing and its Applications* (2001), vol. 1, IEEE, pp. 13–16.

[13] Bousbia-Salah, H., Belouchrani, A., and Abed-Meraim, K. Jacobi-like algorithm for blind signal separation of convolutive mixtures. *Electronics Letters 37*, 16 (2001), 1049–1050.

[14] Boussé, M., Debals, O., and De Lathauwer, L. A tensor-based method for large-scale blind source separation using segmentation. *IEEE Trans. Signal Process. 65*, 2 (2017), 346–358.

[15] Bro, R. PARAFAC. Tutorial and applications. *Chemometr. Intell. Lab. 38*, 2 (1997), 149–171.

[16] Bro, R., Harshman, R. A., Sidiropoulos, N. D., and Lundy, M. E. Modeling multi-way data with linearly dependent loadings. *Journal of Chemometrics 23*, 7-8 (2009), 324–340.

[17] Caiafa, C. F., and Cichocki, A. Multidimensional compressed sensing and their applications. *WIREs Data Mining Knowl. Discov. 3*, 6 (2013), 355–380.

[18] Cardoso, J.-F. Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors. In *International Conference on Acoustics, Speech, and Signal Processing, 1991. ICASSP-91.* (1991), IEEE, pp. 3109–3112.

[19] Cardoso, J.-F., Bose, S., and Friedlander, B. Output cumulant matching for source separation. In *Proc. IEEE Signal Processing / ATHOS Workshop on Higher-Order Statistics, Girona, Spain* (1995), pp. 44–50.

[20] Cardoso, J.-F., Bose, S., and Friedlander, B. On optimal source separation based on second and fourth order cumulants. In *Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing* (1996), IEEE, pp. 198–201.

[21] Cardoso, J.-F., et al. On the performance of orthogonal source separation algorithms. In *Proceedings of the IEEE European Signal Processing Conference (EUSIPCO)* (1994), vol. 94, Edinburgh, UK, pp. 776–779.

[22] Cardoso, J.-F., and Souloumiac, A. Blind beamforming for non-Gaussian signals. In *Radar and Signal Processing, IEE Proc.-F* (1993), vol. 140, IET, pp. 362–370.

[23] C.E.R. Fernandes, Favier, G., and J.C.M. Mota. Blind channel identification algorithms based on the parafac decomposition of cumulant tensors: The single user and multiuser cases. *Signal Processing 88* (2008), 1382–1401.

[24] Chen, B., and Petropulu, A. P. Frequency domain blind MIMO system identification based on second-and higher order statistics. *IEEE Trans. Signal Process. 49*, 8 (2001), 1677–1688.

[25] Cichocki, A., Mandic, C., Phan, A. H., Caifa, C., Zhou, G., Zhao, Q., and De Lathauwer, L. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag. 32*, 2 (2015), 145–163.

[26] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[27] Comon, P. Independent Component Analysis, a new concept ? *Signal Processing 36* (1994), 287–314.

[28] Comon, P., and Jutten, C., Eds. *Handbook of blind source separation, independent component analysis and applications.* Academic Press, 2010.

[29] De Lathauwer, L. *Signal processing based on multilinear algebra.* Ph.d. dissertation, KU Leuven, 1997.

[30] De Lathauwer, L. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM J. Matrix Anal. Appl. 28*, 3 (2006), 642–666.

[31] De Lathauwer, L. Decomposition of a higher-order tensor in block terms — Part II: Definitions and uniqueness. *SIAM J. Matrix Anal. Appl. 30* (2008), 1033–1066.

[32] De Lathauwer, L. Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(L_r, L_r, 1)$ terms. *SIAM J. Matrix Anal. Appl. 32*, 4 (2011), 1451–1474.

[33] De Lathauwer, L. Characterizing higher-order tensors by means of subspaces. tech. report 11–32, ESAT–STADIUS, KU Leuven, Belgium, 2011.

[34] De Lathauwer, L., and Castaing, J. Blind Identification of Underdetermined Mixtures by Simultaneous Matrix Diagonalization. *IEEE Trans. Signal Process. 56*, 3 (2008), 1096–1105.

[35] De Lathauwer, L., De Moor, B., and Vandewalle, J. Independent component analysis based on higher-order statistics only. In *Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing* (1996), IEEE, pp. 356–359.

[36] De Lathauwer, L., De Moor, B., and Vandewalle, J. Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Trans. Biomed. Eng. 47*, 5 (2000), 567–572.

[37] De Lathauwer, L., De Moor, B., and Vandewalle, J. An introduction to independent component analysis. *Journal of chemometrics 14*, 3 (2000), 123–149.

[38] De Lathauwer, L., De Moor, B., and Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl. 21*, 4 (2000), 1253–1278.

[39] De Lathauwer, L., De Moor, B., and Vandewalle, J. On the Best Rank-1 and Rank-(R_1, R_2, ..., R_N) Approximation of Higher Order Tensors. *SIAM J. Matrix Anal. Appl. 21*, 4 (2000), 1324–1342.

[40] De Lathauwer, L., De Moor, B., and Vandewalle, J. A prewhitening-induced bound on the identification error in independent component analysis. *IEEE Trans. Circuits Syst. I 52*, 3 (2005), 546–554.

[41] De Lathauwer, L., and Nion, D. Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms. *SIAM J. Matrix Anal. Appl. 30*, 3 (2008), 1067–1083.

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

[43] De Moor, B. The singular value decomposition and long and short spaces of noisy matrices. *IEEE Trans. Signal Process. 41*, 9 (1993), 2826–2838.

[44] Debals, O., and De Lathauwer, L. Stochastic and deterministic tensorization for blind signal separation. *International Conference on Latent Variable Analysis and Signal Separation* (2015), 3–13.

[45] Debals, O., and De Lathauwer, L. The concept of tensorization. Tech. Rep. 17-99, ESAT-STADIUS, KU Leuven, Belgium, 2017.

[46] Debals, O., Van Barel, M., and De Lathauwer, L. Löwner-based blind signal separation of rational functions with applications. *IEEE Trans. Signal Process. 64*, 8 (2016), 1909–1918.

[47] Debals, O., Van Eeghem, F., Vervliet, N., and De Lathauwer, L. Tensorlab Demos - Release 3.0. Tech. Rep. 16-68, ESAT-STADIUS, KU Leuven, Belgium, 2016.

[48] DeGroat, R., Dowling, E., and Linebarger, D. Subspace tracking. In *The Digital Signal Processing Handbook*, V. Madisetti, Ed. CRC Press, 1998.

[49] Domanov, I., and De Lathauwer, L. On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part I: Basic results and uniqueness of one factor matrix. *SIAM J. Matrix Anal. Appl. 34*, 3 (2013), 855–875.

[50] Domanov, I., and De Lathauwer, L. On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part II: Overall uniqueness. *SIAM J. Matrix Anal. Appl. 34*, 3 (2013), 876–903.

[51] Domanov, I., and De Lathauwer, L. Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition. *SIAM J. Matrix Anal. Appl. 35*, 2 (2014), 636–660.

[52] Domanov, I., and De Lathauwer, L. Canonical polyadic decomposition of third-order tensors: relaxed uniqueness conditions and algebraic algorithm. *Linear Algebra Appl. 513* (2017), 342–375.

[53] Domanov, I., and De Lathauwer, L. From computation to comparison of tensor decompositions. *arXiv:1912.04694* (2019).

[54] Domanov, I., and De Lathauwer, L. On uniqueness and computation of the decomposition of a tensor into multilinear rank-$(1, L_r, L_r)$ terms. *SIAM J. Matrix Anal. Appl. 41*, 2 (2020), 747–803.

[55] Domanov, I., Vervliet, N., and De Lathauwer, L. Decomposition of a tensor into multilinear rank $(M_r, N_r, \cdot)$ terms. Tech. Rep. 18-51, ESAT-STADIUS, KU Leuven, Belgium, 2018.

[56] Dreesen, P., Westwick, D., Schoukens, J., and Ishteva, M. Modeling parallel Wiener-Hammerstein systems using tensor decomposition of Volterra kernels. In *13th International Conference on*

Latent Variable Analysis and Signal Separation (LVA/ICA) (Feb. 2017), Springer, pp. 16–25.

[57] Favier, G., and Kibangou, A. Tensor-based methods for system identification. Part 2: Three examples of tensor-based system identification methods. *International Journal on Sciences and Techniques of Automatic Control (IJ-STA) 3*, 1 (2009), 870–889.

[58] Fernandes, C. E. R., Favier, G., and Mota, J. C. M. Blind multipath MIMO channel parameter estimation using the Parafac decomposition. In *IEEE International Conference on Communications, 2009. ICC'09.* (2009), IEEE, pp. 1–5.

[59] Fernandes, C. E. R., Favier, G., and Mota, J. C. M. PARAFAC-based blind identification of convolutive MIMO linear systems. In *IFAC Proceedings Volumes* (2009), vol. 42, Elsevier, pp. 1704–1709.

[60] Gelle, G., Colas, M., and Serviere, C. Blind source separation: a tool for rotating machine monitoring by vibrations analysis? *Journal of sound and vibration 248*, 5 (2001), 865–885.

[61] Gorokhov, A., and Loubaton, P. Subspace-based techniques for blind separation of convolutive mixtures with temporally correlated sources. *IEEE Trans. Circuits Syst. I 44*, 9 (1997), 813–820.

[62] Goulart, J. H. M., Boizard, M., Boyer, R., Favier, G., and Comon, P. Tensor CP decomposition with structured factor matrices: Algorithms and performance. *IEEE J. Sel. Topics Signal Process.* (To appear).

[63] Goulart, J. H. M., and Favier, G. An algebraic solution for the Candecomp/Parafac decomposition with circulant factors. *SIAM J. Matrix Anal. Appl. 35*, 4 (2014), 1543–1562.

[64] Grasedyck, L., Kressner, D., and Tobler, C. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen 36*, 1 (2013), 53–78.

[65] Gunning, R. C., and Rossi, H. *Analytic functions of several complex variables*, vol. 368. American Mathematical Soc., 2009.

[66] Håstad, J. Tensor rank is NP-complete. *Journal of Algorithms 11*, 4 (1990), 644–654.

[67] Hillar, C. J., and Lim, L.-H. Most tensor problems are NP-hard. *Journal of the ACM (JACM) 60*, 6 (2013), 45.

[68] Hsu, D., and Kakade, S. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science* (2013), pp. 11–20.

[69] Huang, F., and Anandkumar, A. Convolutional dictionary learning through tensor factorization. In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges, NIPS* (2015), pp. 116–129.

[70] Hyvärinen, A., Karhunen, J., and Oja, E. *Independent component analysis*. John Wiley & Sons, 2004.

[71] Hyvärinen, A., and Oja, E. Independent component analysis: algorithms and applications. *Neural networks 13*, 4 (2000), 411–430.

[72] Icart, S., Comon, P., and Rota, L. Blind paraunitary equalization. *Signal process. 89*, 3 (2009), 283–290.

[73] James, C. J., and Hesse, C. W. Independent component analysis for biomedical signals. *Physiological measurement 26*, 1 (2005), R15–R39.

[74] Jiang, T., and Sidiropoulos, N. D. Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear model with constant modulus constraints. *IEEE Trans. Signal Process. 52*, 9 (Sept. 2004), 2625–2636.

[75] Kang, U., Papalexakis, E., Harpale, A., and Faloutsos, C. 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* (2012), ACM, pp. 316–324.

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

[77] Kibangou, A., and Favier, G. Wiener-Hammerstein systems modeling using diagonal Volterra kernels coefficients. *IEEE Signal Proc. Letters 13*, 6 (June 2006), 381–384.

[78] Kibangou, A. Y., and Favier, G. Non-Iterative Solution for PARAFAC with a Toeplitz Matrix Factor. In *Proc. EUSIPCO* (Aug. 24-28, Glasgow, Scotland, 2009).

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

[80] Knyazev, A. V., and Argentati, M. E. Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing 23*, 6 (2002), 2008–2040.

[81] Kolda, T. G., and Bader, B. W. Tensor decompositions and applications. *SIAM Rev. 51*, 3 (2009), 455–500.

[82] Krim, H., and Viberg, M. Two decades of array signal processing research: the parametric approach. *IEEE Signal Process. Mag. 13*, 4 (1996), 67–94.

[83] Kroonenberg, P. M. *Applied multiway data analysis.* John Wiley & Sons, 2008.

[84] Kruskal, J. B. Three-way arrays: Rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics. *Linear Algebra Appl. 18* (1977), 95–138.

[85] Lahat, D., Adali, T., and Jutten, C. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proc. IEEE 103*, 9 (2015), 1449–1477.

[86] Leurgans, S., and Ross, R. Multilinear models: applications in spectroscopy. *Statistical Science* (1992), 289–310.

[87] Leurgans, S., Ross, R., and Abel, R. A decomposition for three-way arrays. *SIAM J. Matrix Anal. Appl. 14*, 4 (1993), 1064–1083.

[88] Liang, J., and Ding, Z. Blind MIMO system identification based on cumulant subspace decomposition. *IEEE Trans. Signal Process. 51*, 6 (2003), 1457–1468.

[89] Liao, T. Clustering of time series data — a survey. *Pattern recognition 38*, 11 (2005), 1857–1874.

[90] Liu, H., and Xu, G. Closed-form blind symbol estimation in digital communication. *IEEE Trans. Signal Process. 43*, 11 (1995), 2714–2723.

[91] Liu, H., and Xu, G. Multiuser blind channel estimation and spatial channel pre-equalization. In *International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95* (1995), vol. 3, IEEE, pp. 1756–1759.

[92] Liu, H., and Xu, G. Smart antennas in wireless systems: uplink multiuser blind channel and sequence detection. *IEEE Trans. Communications 45*, 2 (1997), 187–199.

[93] Mansour, A., Barros, A. K., and Ohnishi, N. Blind separation of sources: Methods, assumptions and applications. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 83*, 8 (2000), 1498–1512.

[94] Mansour, A., Jutten, C., and Loubaton, P. Subspace method for blind separation of sources in convolutive mixture. In *Proc. EUSIPCO* (1996), pp. 2081–2084.

[95] Markovsky, I. *Low rank approximation: algorithms, implementation, applications*. Springer, 2012.

[96] Marshall, P., Rajguru, N., and Slosar, A. Bayesian evidence as a tool for comparing datasets. *Physical Review D 73*, 6 (2006).

[97] Meilă, M. Comparing clusterings: an information based distance. *Journal of multivariate analysis 98*, 5 (2007), 873–895.

[98] Moulines, E., Cardoso, J.-F., and Gassiat, E. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (1997), vol. 5, IEEE, pp. 3617–3620.

[99] Nion, D., and Sidiropoulos, N. D. Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radar. *IEEE Trans. Signal Process. 58*, 11 (2010), 5693–5705.

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

[101] Papalexakis, E., Faloutsos, C., and Sidiropoulos, N. Parcube: Sparse parallelizable tensor decompositions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2012), Springer, pp. 521–536.

[102] Pearson, R. Using many models to compare datasets, 2018. Available online at https://cran.r-project.org/web/packages/datarobot/vignettes/ComparingSubsets.html.

[103] Pedersen, M., Larsen, J., Kjems, U., and Parra, L. A survey of convolutive blind source separation methods. *Multichannel Speech Processing Handbook* (2007), 1065–1084.

[104] Phan, A. H., Tichavský, P., and Cichocki, A. Low complexity damped Gauss–Newton algorithms for CANDECOMP/PARAFAC. *SIAM J. Matrix Anal. Appl. 34*, 1 (2013), 126–147.

[105] Roy, R., and Kailath, T. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing 37*, 7 (1989), 984–995.

[106] Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation 34*, 3 (1986), 276–280.

[107] Sidiropoulos, N., Papalexakis, E., and Faloutsos, C. A parallel algorithm for big tensor decomposition using randomly compressed cubes (paracomp). In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2014)* (2014), IEEE, pp. 1–5.

[108] Sidiropoulos, N. D., and Bro, R. On the Uniqueness of Multilinear Decomposition of N-way Arrays. *J. Chemometrics 14* (2000), 229–239.

[109] Sidiropoulos, N. D., Bro, R., and Giannakis, G. B. Parallel factor analysis in sensor array processing. *IEEE Trans. Signal Process. 48*, 8 (2000), 2377–2388.

[110] Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process. 65*, 13 (2017), 3551–3582.

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

[112] Smilde, A., Bro, R., and Geladi, P. *Multi-way analysis: applications in the chemical sciences.* John Wiley & Sons, 2005.

[113] Sorber, L., Van Barel, M., and De Lathauwer, L. Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization. *SIAM J. Optim. 23*, 2 (2013), 695–720.

[114] Sorber, L., Van Barel, M., and De Lathauwer, L. Structured data fusion. *IEEE J. Sel. Topics Signal Process. 9*, 4 (2015), 586–600.

[115] Sørensen, M., and Comon, P. Tensor decompositions with banded matrix factors. *Linear Algebra Appl. 438*, 2 (2013), 919–941.

[116] Sørensen, M., and De Lathauwer, L. Blind signal separation via tensor decomposition with Vandermonde factor: Canonical polyadic decomposition. *IEEE Trans. Signal Process. 61*, 22 (2013), 5507–5519.

[117] Sørensen, M., and De Lathauwer, L. Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part I: Uniqueness. *SIAM J. Matrix Anal. Appl. 36*, 2 (2015), 496–522.

[118] Sørensen, M., and De Lathauwer, L. New uniqueness conditions for the canonical polyadic decomposition of third-order tensors. *SIAM J. Matrix Anal. Appl. 36*, 4 (2015), 1381–1403.

[119] Sørensen, M., and De Lathauwer, L. Tensor decompositions with a block-Toeplitz factor matrix. Tech. Rep. 11-215, ESAT-STADIUS, KU Leuven, Belgium, 2016.

[120] Sørensen, M., and De Lathauwer, L. Fiber sampling approach to canonical polyadic decomposition and application to tensor completion. *SIAM J. Matrix Anal. Appl. 40*, 3 (2019), 888–917.

[121] Sørensen, M., De Lathauwer, L., and Comon, P. Fourth-order Cumulant Tensor Based Identification of Underdetermined MIMO FIR Channels. Tech. Rep. 20-113, ESAT-STADIUS, KU Leuven, Belgium, 2011.

[122] Sørensen, M., De Lathauwer, L., Icart, S., and Deneire, L. On Jacobi-type methods for blind equalization of paraunitary channels. *Signal process. 92*, 3 (2012), 617–624.

[123] Sørensen, M., Domanov, I., and De Lathauwer, L. Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part II: Algorithms. *SIAM J. Matrix Anal. Appl. 36*, 3 (2015), 1015–1045.

[124] Sorensen, M., Van Eeghem, F., and De Lathauwer, L. Blind multichannel deconvolution and convolutive extensions of canonical polyadic and block term decompositions. *IEEE Trans. Signal Process. 65*, 15 (2017), 4132–4145.

[125] Stoica, P., and Sharman, K. C. Maximum likelihood methods for direction-of-arrival estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing 38*, 7 (1990), 1132–1143.

[126] Tandon, N., and Choudhury, A. A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology international 32*, 8 (1999), 469–480.

[127] Tichavský, P., and Koldovsky, Z. Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources. *IEEE Trans. Signal Process. 59*, 3 (2011), 1037–1047.

[128] Tichavský, P., and Yeredor, A. Fast approximate joint diagonalization incorporating weight matrices. *IEEE Trans. Signal Process. 57*, 3 (2009), 878–891.

[129] Tomasi, G., and Bro, R. Parafac and missing values. *Chemometr. Intell. Lab. 75*, 2 (2005), 163–180.

[130] Tomasi, G., and Bro, R. A comparison of algorithms for fitting the PARAFAC model. *Computational Statistics & Data Analysis 50*, 7 (2006), 1700–1734.

[131] Tong, L. Identification of multichannel MA parameters using higher-order statistics. *Signal Processing 53* (1996), 195–209.

[132] Tong, L., and Perreau, S. Multichannel blind identification: From subspace to maximum likelihood methods. *Proc. IEEE 86* (1998), 1951–1968.

[133] Treichler, J., and Agee, B. A new approach to multipath correction of constant modulus signals. *IEEE Trans. Acoustics, Speech and Signal Processing 31*, 2 (1983), 459–472.

[134] Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika 31*, 3 (1966), 279–311.

[135] Tucker, L. R., et al. The extension of factor analysis to three-dimensional matrices. *Contributions to mathematical psychology* (1964).

[136] Van Der Veen, A.-J. Joint diagonalization via subspace fitting techniques. In *Proc. ICASSP* (Salt Lake City, USA, 07-11 2001).

[137] Van Der Veen, A.-J., and Paulraj, A. An analytical constant modulus algorithm. *IEEE Trans. Signal Process. 44*, 5 (1996), 1136–1155.

[138] Van der Veen, A.-J., Talwar, S., and Paulraj, A. Blind estimation of multiple digital signals transmitted over FIR channels. *IEEE Signal Process. Lett. 2*, 5 (1995), 99–102.

[139] Van Der Veen, A.-J., Talwar, S., and Paulraj, A. A subspace approach to blind space-time signal processing for wireless communication systems. *IEEE Trans. Signal Process. 45*, 1 (1997), 173–190.

[140] Van Eeghem, F., and De Lathauwer, L. Blind system identification as a compressed sensing problem. Tech. Rep. 15-170, ESAT-STADIUS, KU Leuven, Belgium, 2015.

[141] Van Eeghem, F., and De Lathauwer, L. Second-order tensor-based convolutive ICA: deconvolution versus tensorization. In *Proc. of the 42th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), pp. 2252–2256.

[142] Van Eeghem, F., and De Lathauwer, L. Algorithms for canonical polyadic decompositions with block-circulant factors. *IEEE Signal Process. Lett. 25*, 6 (2018), 798–802.

[143] Van Eeghem, F., and De Lathauwer, L. Determining whether two datasets cluster similarly without determining the clusters. Tech. Rep. 17-81, ESAT-STADIUS, KU Leuven, Belgium, 2020.

[144] Van Eeghem, F., and De Lathauwer, L. Tensor similarity in chemometrics. In *Comprehensive Chemometrics: Chemistry, Molecular Sciences and Chemical Engineering*, S. Brown, R. Tauler, and B. Walczak, Eds. Elsevier, 2020, pp. 337–354.

[145] Van Eeghem, F., Debals, O., and De Lathauwer, L. Tensor similarity in two modes. *IEEE Trans. Signal Process. 66*, 5 (2018), 1273–1285.

[146] Van Eeghem, F., Debals, O., Vervliet, N., and De Lathauwer, L. Coupled and incomplete tensors in blind system identification. *IEEE Trans. Signal Process. 66*, 23 (2018), 6137–6147.

[147] Van Eeghem, F., Sørensen, M., and De Lathauwer, L. Tensor decompositions with several block-Hankel factors and application in blind system identification. *IEEE Trans. Signal Process. 65*, 15 (2017), 4090–4101.

[148] Vandecappelle, M., Bousse, M., Van Eeghem, F., and De Lathauwer, L. Tensor decompositions for graph clustering. Tech. Rep. 16-170, ESAT-STADIUS, KU Leuven, Belgium, 2016.

[149] Vasilescu, M. A. O., and Terzopoulos, D. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision* (2002), Springer, pp. 447–460.

[150] Vervliet, N., and De Lathauwer, L. A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors. *IEEE J. Sel. Topics Signal Process. 10*, 2 (2016), 284–295.

[151] Vervliet, N., and De Lathauwer, L. Numerical optimization-based algorithms for data fusion. *Data Handling in Science and Technology 33* (2019), 81–128.

[152] Vervliet, N., Debals, O., and De Lathauwer, L. Canonical polyadic decomposition of incomplete tensors with linearly constrained factors. Tech. Rep. 16-172, ESAT-STADIUS, KU Leuven, Belgium, 2016.

[153] Vervliet, N., Debals, O., and De Lathauwer, L. Tensorlab 3.0 — numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In *50th Asilomar Conference on Signals, Systems and Computers* (2016), IEEE, pp. 1733–1738.

[154] Vervliet, N., Debals, O., and De Lathauwer, L. Exploiting efficient representations in large-scale tensor decompositions. *SIAM Journal on Scientific Computing 41*, 2 (2019), A789–A815.

[155] Vervliet, N., Debals, O., Sorber, L., and De Lathauwer, L. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Process. Mag. 31*, 5 (2014), 71–79.

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

[157] Vervliet, N., Vandecappelle, M., Bousse, M., Zink, R., and De Lathauwer, L. Recent numerical and conceptual advances for tensor decompositions — A preview of Tensorlab 4.0. In *Proc. of the 2019 IEEE Data Science Workshop (IEEE DSW)* (2019), pp. 310–314.

[158] Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research 11*, Oct (2010), 2837–2854.

[159] Wagner, S., and Wagner, D. *Comparing clusterings: an overview.* Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[160] Wiener, J., and Bronson, N. Facebook's top open data problems, October 2014. [Online; visited Sep. 18, 2018].

[161] Xu, R., and Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on neural networks 16*, 3 (2005), 645–678.

[162] Yang, B. Projection approximation subspace tracking. *IEEE Trans. Signal Process. 43*, 1 (1995), 95–107.

[163] Yeredor, A. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Trans. Signal Process. 50*, 7 (2002), 1545–1553.

[164] Yu, Y., and Petropulu, A. PARAFAC-based blind estimation of possibly underdetermined convolutive MIMO systems. *IEEE Trans. Signal Process. 56*, 1 (2008), 111–124.

[165] Zarzoso, V., Nandi, A. K., Igual-Garcia, J., and Vegara-Dominguez, L. Blind identification and equalization of MIMO FIR channels based on subspace decomposition and independent component analysis. In *Proc. 2nd IMA Intl. Conf. on Mathematics in Communications* (Dec. 16-18, Lancaster, UK, 2002).

[166] Zibulevsky, M., and Pearlmutter, B. A. Blind source separation by sparse decomposition in a signal dictionary. *Neural computation 13*, 4 (2001), 863–882.

Curriculum Vitae

Frederik Van Eeghem graduated as Master of Science in Mathematical Engineering with greatest distinction at KU Leuven in 2014. After this, Frederik started pursuing a PhD in tensor-based signal processing at the STADIUS Center of Dynamical Systems, Signal Processing and Data Analytics of the Electrical Engineering Department of KU Leuven under the guidance of Prof. Lieven De Lathauwer. The focus of his research lies in tensor methods for blind system identification and independent component analysis, as well as tensor similarity.

Since 2017, Frederik has combined his PhD research with a full-time job as a technology consultant, where he focuses on the boundary between strategy and IT and specializes in machine learning. He has realized projects for European institutions, universities, and international private companies in telecommunications, finance, retail and technology.

List of publications

Articles in peer-reviewed journals

[1] Van Eeghem, F., Debals, O., and De Lathauwer, L. Tensor similarity in two modes. *IEEE Trans. Signal Process. 66*, 5 (2018), 1273–1285

[2] Van Eeghem, F., Sørensen, M., and De Lathauwer, L. Tensor decompositions with several block-Hankel factors and application in blind system identification. *IEEE Trans. Signal Process. 65*, 15 (2017), 4090–4101

[3] Van Eeghem, F., and De Lathauwer, L. Algorithms for canonical polyadic decompositions with block-circulant factors. *IEEE Signal Process. Lett. 25*, 6 (2018), 798–802

[4] Van Eeghem, F., Debals, O., Vervliet, N., and De Lathauwer, L. Coupled and incomplete tensors in blind system identification. *IEEE Trans. Signal Process. 66*, 23 (2018), 6137–6147

[5] Sorensen, M., Van Eeghem, F., and De Lathauwer, L. Blind multichannel deconvolution and convolutive extensions of canonical polyadic and block term decompositions. *IEEE Trans. Signal Process. 65*, 15 (2017), 4132–4145

Articles in peer-reviewed conference proceedings

[1] Van Eeghem, F., and De Lathauwer, L. Second-order tensor-based convolutive ICA: deconvolution versus tensorization. In *Proc. of the 42th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), pp. 2252–2256

Book chapters

[1] Van Eeghem, F., and De Lathauwer, L. Tensor similarity in chemometrics. In *Comprehensive Chemometrics: Chemistry, Molecular Sciences and Chemical Engineering*, S. Brown, R. Tauler, and B. Walczak, Eds. Elsevier, 2020, pp. 337–354

Reports

[1] Debals, O., Van Eeghem, F., Vervliet, N., and De Lathauwer, L. Tensorlab Demos - Release 3.0. Tech. Rep. 16-68, ESAT-STADIUS, KU Leuven, Belgium, 2016

[2] Vandecappelle, M., Bousse, M., Van Eeghem, F., and De Lathauwer, L. Tensor decompositions for graph clustering. Tech. Rep. 16-170, ESAT-STADIUS, KU Leuven, Belgium, 2016

[3] Van Eeghem, F., and De Lathauwer, L. Blind system identification as a compressed sensing problem. Tech. Rep. 15-170, ESAT-STADIUS, KU Leuven, Belgium, 2015

[4] Van Eeghem, F., and De Lathauwer, L. Determining whether two datasets cluster similarly without determining the clusters. Tech. Rep. 17-81, ESAT-STADIUS, KU Leuven, Belgium, 2020



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING, AND DATA ANALYTICS
Kasteelpark Arenberg 10 bus 2440
B-3001 Leuven
frederik.vaneeghem@kuleuven.be
https://www.esat.kuleuven.be/stadius/